



November 1988

Vol. 2

Nº 2

Price £1.80

# Archive

*The Subscription Magazine for Archimedes Users*



Fractal Landscapes

---

RiscForth Review

---

First Word Plus Printer Drivers

---

Textdump Module

Image Processing / Toolkit Plus Reviews

WIMP Templates / Checksum Routine

ISO Pascal Extensions / Arthur – 'C' Clerk

---



## Under seige

"While I'm sending in my subscription, I'll just write and tell Paul/ ask him /send him this program...". Thanks very much for all the material you've sent in, but there is so much of it that I'm a bit swamped! I hope you'll bear with us if we're a little slower than usual at replying to your queries.

Also, it would be really helpful, if you feel your contribution might be publishable, even in part, to send the letter on a 640k disc as well as on paper. Thanks.

*(Yes, 640k, please because it's still easier for me to use the Master with a 3.5" drive to link up to the Mac rather than using the Archimedes.)*

## Pig in the middle

It's quite fascinating, but sometimes rather frustrating, to stand in the middle between the hardware and software producers on the one hand and the consumers on the other. The nice part of it is that there are some really helpful and generous business people who genuinely do their best to help their customers and there are some extremely helpful users who really want to help the producers improve the quality of their service.

On the other hand, there are some companies who are convinced they are giving the best service in the world and then, as I try to get them to see things from the customers' viewpoint a little more, they think I am trying to damage their reputation by what I say in the magazine!

The producers don't have the monopoly of unreasonableness! They have to put up, for example, with the customer who **has** to have the latest version of the software – he's got version 1.234 but he has read that there is a version 1.235 so he expects a free up-grade – and this is not because he's actually found the minor 'features' that caused the company to do the new version. In his eyes, his version must have some sort of bug, so it is deficient. So, having paid good money for it, a bug-free version is his right! But what if the up-grade was to improve the facilities, should he still demand his free up-grade?

Then there's the customer who insists that there's something wrong with the program because it won't do something or other. The company representative tries to explain that it's something wrong with the way it is being used or a clash with some other software because hundreds of others are using it quite successfully. Then, after several long and expensive phone calls, the customer demands his money back and threatens legal action as an alternative.

You may think I'm exaggerating, but I'm not. There are people around like this – on both sides! If you recognise yourself in any of these comments then I do hope that you will try to see things from the other side a bit more. Surely, it is in the customer's interests in the long run that the producing companies expand and succeed – they get more choice of software. From the other direction, if the customer is generally happy that the company is trying to be helpful he's going to buy more product.

From my point of view, I have to tread the fine line between the two sides – to filter out the unreasonable criticism and get a constructive dialogue going. If I seem to be allowing Archive to contain unreasonable criticism, you should see the comments I didn't publish! I try to be fair to everyone and if anyone feels hard done by, please say so and be patient with me in my role as 'pig in the middle'!

Thanks for all your help, guys!



---

# Archive

---

Volume 2 • Nº 2 • November 1988

---

## Contents

Hardware & Software Available ...2	Toolkit Plus Review .....32
Readers' Comments .....3	MS-DOS Column .....33
Hints & Tips .....6	Assembly Programming – 5 .....35
Small Ad's .....14	WIMP Templates .....36
Fractal Landscapes .....17	BASIC Forum – Checksums.....39
Help – Needed and Offered .....20	ISO Pascal Extensions – 2 .....41
Contact Box .....21	Arthur – 'C' Clerk.....43
Who will venture Forth? .....23	ASCII, Printer Drivers & FWP...47
Matters Arising .....26	Image Processing from Holland .52
A Textdump Module .....27	Fact-File .....53
Hardware Column .....30	



## Hardware & Software Available

• **Repton3** – yes, it is being done (in native mode) for the Archimedes, and it should be released by Superior Software at the end of October, i.e. it should have been released by the time you read this!

• **Inter Series on Disc!!!** – Computer Concepts have bowed to pressure from various quarters and are now selling Inter-Word, Inter-Sheet, Inter-Chart and Spell-Master on disc. (£29, £29, £19 and £39 respectively, all + VAT, cf £39, £39, £25 and £51.30 for the ROM versions.) They still work under the emulator and there is no ROM-LINK facility as there is on the BBC micro versions. (Archive prices are £30, £30, £20 and £41 inclusive of VAT + p&p.)

• **Desktop Enhancer Enhanced!** – The new version has options to copy read only files during backup and to boot floppy disc from the FDD icon menu. Mitre Software have also added a utility to change the machine configuration without <ctrl-break>. Existing users can return their discs with an S.A.E. for a free up-grade. (Other software houses, please note!) (Available through Archive for £27.)

• **Pro-Artisan** – Should be available from Clares Micros soon though at the time of going to press, we're still waiting. "Probably mid November..."

• **Artisan Gallery Discs** – Clares now have three discs at £6 each or (£15 for the set) each containing nine Artisan pictures. (Only available from Clares as no dealer discount is available.)

• **Pipedream SpellCheck** should be available from Colton Software at the beginning of November, price £49.45 (£45 through Archive.)

• **Colour Convertor** – Lingenuity's single width podule (£169.95 +VAT) – converts the Watford digitiser into full colour! Hardware controls of both contrast and saturation of the separate colours. Controlling software is mouse driven and allows you to save a picture, scale it or set the parameters for the hardware controls. They hope it will be available by 11th November. (£185 from Archive)

• **Control Panel** (£14.95 + VAT) – Lingenuity are also producing a WIMP driven control panel with on-screen help which will allow you to set up the

configuration of your machine including loading standard configurations from disc for use with your various applications. (£16 from Archive)

• **Xample Story** – the presentation graphics package from Holland, mentioned in last month's review from the PC Show (page 7) is now being distributed by Lingenuity.

• **AIM the Image processing package** from Delft University, mentioned in the September issue is now also being distributed by Lingenuity. They are providing the disc and the documentation (see the review on page 52) at cost price which will be "about £6" and this includes a pound which will be sent to the Delft University students.

• **BASIC Compilers** – the two new BASIC compilers, one from Dabs Press at £99.95 inclusive and the other from Silicon Vision also at £99.95 should both be ready by the time you read this. We hope to have a comparative review next month. (Both will be available through Archive at £92.)

• **Links to Psion Organiser** – XOB have written ARCPERX which runs under the 6502 emulator and uses the 1MHz bus on the I/O podule to link up to a Psion Organiser.

• **DFS utilities** – CJE Micros do a DFS utilities disc for £20 inclusive which allows you to read and write and format DFS discs from the Archimedes – it will even read dual catalogue discs.

• **Multi-syncs galore** – Solidisk Technology have several multi-sync monitors for the Archimedes. If you want colour, there's the Quadram at £300 or the TVM/MD11 at £329 but if you want the high resolution but aren't worried about having colour, they have two grey scale multi-syncs, NEC and TVM, both £159. All the monitors mentioned are 14" and to all the prices you need to add VAT and carriage. (I haven't seen any of them 'in the flesh', so we'd welcome comments from anyone who has.)

• **Relational Database** – 'Reporter' (£39.95) from Minerva Systems extends the power of System Delta Plus – a fully relational report-writer with averaging, totalling etc. Should be available by the time you get this magazine. (£37 through Archive)



• **Mailshot & Label printing** – 'Mailshot' (£39.95) from Minerva Systems also extends the power of System Delta Plus – print sticky labels or do your mailmerge from your database, "advanced date handling includes conversion of numbers to words!" (£37 through Archive)

• **Drawing Board** – £59 inc. Another innovative program from Jansons who produced ArcImEd, the art package that generates program listings. It's a drawing package designed for dot matrix printers. On demonstration it looked very impressive. See the advert on page 9 for a sample of the output.

• **Two new games** from GEM Electronics – Word Up Word Down – "a new concept in word games featuring a 3D game board"! and Startrader, "a large scale space adventure game". Each is £17.95 inclusive and if you buy them both, you get their Desktop Games (£5.95) free!

• **Arckey** – a keystrip flip-over holder from Ian Copestake – self adhesive clear plastic allows you to stick your keystrips together into a 'book'.

• **Crossword puzzles** – you can now get puzzles

from the Times and the Sun newspapers on computer with help offered if you need it. Akom Ltd are offering Times puzzles for £16.95 and Sun puzzles for £15.95. (Also available for BBC micro.)

• **New Archimedes bulletin board** – if you have got tired of waiting for Eureka!, why not give ARC-port a ring on 0252-737065, 300/300 or 1200/75, 8n1. It's run on an Archimedes and is open Mon – Friday 6.00 p.m. to 8.00 a.m. + 24 hrs at weekends.

• **New Archive Shareware Disc** – (though I think, according to Ken Biddle's definition, we should call it PD! Oops!) Shareware Disc 2 will be ready by the time you read this. £3 as usual. I'll try to put more details on the Order Form of what is on it.

### Review Software Received...

Apart from reviews already written we have received review copies of the following software: Ovid Toolkit Module, Akom Crossword puzzles, Cobra (like the BBC snake game that eats things and gets longer) from Richard Millican, Plague Planet from Alpine Software, Shares and Accounts from Ian Hamilton. **A**

## Readers' Comments

• **Program listings** – Why is it that I don't find your program listings as easy to type in when compared to your nearest rival? Richard Pepper.

Yes, I'd like to know too. First of all, do others think the same? Secondly is it because of the way long lines are split? RISC User do it by fixing the number of characters to the line and letting it split where it will (which is actually very much easier) whereas I go through each program looking for an 'intelligent' place to split each line, right justifying the remains of the line. Compare the two lines below and see what you think. If the RISC User technique is better, let me know – it will save time.

RISC User style:

```
530 OSCLI("SAVE "+filename$+" "+S
TR$~(workspc%)+""+STR$~(workspc%&
64+&68))
```

Archive style:

```
530 OSCLI("SAVE "+filename$+" "+
STR$~(workspc%)+""+STR$~
(workspc%&64+&68))
```

• **GammaPlot/System Delta Plus** (Ref Archive 2.1, page 9) – I understand from Minerva Software that the latest version number for System Delta Plus is 1.0004 and they expect version 1.0005 "...to be released before the end of the year." SigmaSheet is still in its current release although an upgrade is planned "...again "before the end of the year." Minerva charge £10 (+ VAT) for an upgrade. Upon querying this (and asking if by providing my own disc, the cost would be lower!) I was advised that the cost of upgrades "...is mainly a handling charge..."

(R.A.Brown, Surrey)

*If you need an up-grade to make their two pieces of software compatible then Minerva are happy to provide the up-grade f.o.c. Ed.*

• **ROM/RAM Podules** – A personal view by Brian Cowan – Some months back we had the first critical remarks about ROM/RAM podules and the practice by some software producers of supplying their programs on PLA-ROMs which have to be installed in such a podule. This was welcome news to me; I

thought I must be the only person who was sceptical of this development.

It would have been acceptable (indeed it would have been a very good idea) if the ROMs/RAMs on the board sat directly in the Archimedes' memory space. This would have been the ideal memory expansion scheme. Unfortunately this was not to be; among other problems there is the restriction of the 16 bit podule data bus. Images held on the podule board are loaded over to machine RAM either on power-up or when requested.

So far as I can see, there are three main uses for a ROM/RAM podule. Firstly they can be used to accommodate the above mentioned software. Secondly, any RAM installed in the podule may be used as a RAM filing system. Thirdly, battery backed RAM or custom blown EPROM can hold boot-up initialisation instructions that set the machine up in a specified way when it is switched on. Let's look at these in turn.

**Software ROMs** – The idea of using PLA-ROMs is, of course, to foil the software pirates. However, this also causes problems for bona-fide users. There is the added expense of the podule and the backplane, and also the inconvenience of installation.

Furthermore, this inconvenience is repeated on transferring from one machine to another. I would never buy such software unless it was either (a) far superior to that available on disc or (b) some £100 cheaper than comparable disc based programs (the cost of backplane plus podule).

**RAM Filing system** – It is unfortunate that Arthur version 1.2 does not include a RAM filing system. However the imminent (relatively! Ed.) RISC OS 2 does provide a RAM filing system and this seems the sensible route to me. Another possibility would be for some enterprising programmer or software house to provide an RFS for the existing operating system. I fully understand the problems of those who require the fast access times of a RAM based filing system, but the expense of the backplane plus the podule plus the RAMs to populate it seems excessive. Also, it is not an efficient solution to the problem; a filing system in real RAM address space is the most sensible and must be the fastest implementation. I, therefore, shall exercise patience for the "real thing".

**Boot-up configuring** – The third point, concerning booting up the machine in a particular way, is a little more difficult to answer. If you are lucky enough to have a Winchester installed then that may be configured to perform the required initialisation. Although the cost of this will be outside the reach of many, the purchase of a hard disc must be considered by all serious Archimedes users.

Hopefully prices will fall in the future. In this respect the important consideration is to compare the cost of podule plus complement of RAMs with that of a Winchester system. The other possibility is to use a special "boot-up" disk whenever the machine is turned on. The internal battery backed RAM could be used for simple such applications.

My verdict, then, is a definite "thumbs down" to ROM/RAM podules. However this is an entirely personal view. I am more than willing to be persuaded otherwise. (Brian Cowan, Middlesex)

*Let's hear some other views on this subject. Also please write in if you find that there is any ROM software that is far superior to disc based products. Also, note Computer Concepts' recent decision to release their existing Archimedes software on disc – see Software Available section.*

**• Language Considerations** – I was very interested to read the article by Sean O'Connaill about ArcTFS and especially his comments about the "sneers" because it is written in BASIC.

There does often seem to be a snobbery about languages, often with the implication that the more cryptic the language the better the program. I sometimes feel that at least part of the current enthusiasm for 'C' is the fact that you can write programs which are almost in code! This is nothing new, of course. When I started life as a professional programmer I found that COBOL was despised by many of the senior programmers because "real programmers write in machine code" and there was similar resistance to the use of structure in data names so as to make maintenance easier.

This treatment of COBOL has nothing to do with current criticisms of COBOL as out-of-date, as the only other language available at the time would have been Fortran, which would have been even more unsuitable for the work we were doing.



Part of the problem has, I suspect, come from the fact that "machine-code games" for the early machines were very much better than those written in BASIC because they were much faster. This led to the belief that machine-code programs were automatically better, even though they might be much more difficult to design and amend. This feeling continues into such work as the Dabs Press Archimedes Machine Code book where we are shown how to write machine code to do the work of many of the BASIC constructs.

My own feeling is that I would rather write those things in a language which can be read easily and use machine code for those things which BASIC (or Pascal) finds difficult.

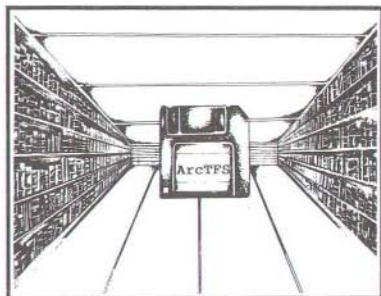
We need to remember that programming languages are for programmers, not the users of the programs. I do not care whether a word-processor has been written in Pascal, C or a language of which I have never heard. It is to be judged on its performance at word-processing – and if it doesn't do that well it is a poor program.

There are, however, disadvantages in using BASIC as a language for programs for professional distribution. Because you have to give the user the source, your ideas are available for anyone to take. If you try to protect yourself by making the program too cryptic, you give yourself a handicap when you have to maintain it. Another consequence is that you may find yourself trying to help someone who is using a program which bears your name but which has been "improved" by somebody else. This has happened to me at work and it was very difficult to help when the listing on the screen didn't match that on my computer.

Most of the other disadvantages of BASIC have disappeared as BASICs have improved and machine memories have expanded. The disadvantages mentioned above will also disappear with the advent of BASIC compilers like that one offered by Dabs Press. I still prefer Pascal, but I certainly wouldn't be stupid enough to sneer at anyone who used BASIC. (David Wild) **A**

## ArcTFS – THE RESEARCH & AUTHOR'S DATABASE

- Indexes, cross-references and files typed or imported ASCII text items up to 7700 characters in length.
- Keeps searchable alphasorted files of explainers for all indexing and source codes.
- Creates additional referencing fields to suit the subject
- Creates text files up to 650K in length on a 800K disc – the equivalent of 80,000–100,000 words.
- Searches text items directly or any referencing field.
- Formats any subset and saves it as a file which will be read as a default – formatted native file, answerable to all reformatting commands, by ArcWriter or 1st Word Plus.
- Translates Wordwise or View files in ASCII form into ArcWriter or 1st Word Plus format, and gives database indexing and control of these files in the meantime.



*"There is no doubt about the value of ArcTFS to the writer or the researcher. It's easy to use, and allows sophisticated searching and sorting of text. Anyone who needs to manage large amounts of this sort of information should find ArcTFS invaluable." says Dave Futcher in Acorn User, November 1988.*

### ArcTFS (Text Filing System) – £29.95

(Cheque or PO only) to:

**TEXCELLENCE, 2 Greenhill Road, Coleraine, N.Ireland BT51 3JE**

(For further details send SAE)



# Hints and Tips

- **Mannisman Tally printers with serial interface** – Pin connections are: pins 1, 4 and 6 linked together on the Archimedes, pin 3 to pin 3 on the printer, pin 5 to pin 7 on the printer and pin 8 to both pins 11 and 19 on the printer.

- **SWI "XOS..."** – If you are using SWI "XOS..." commands so that you can handle your own errors, it will fail if you try to return immediately after this command to the calling routine. A dummy instruction needs to be inserted before the LDMFD instruction. Presumably a 'feature' of OS 1.2.

- **Problems with \*Memory and \*MemoryI** – If you try to use these two commands in the debugger with the aim of dumping to printer, you will find that codes above &7F are not weeded out – they are all sent to the printer regardless which can cause all sorts of problems. The following program patch will solve the problem.

```

10 REM> DEBUGPCH
20 REM (C) JW^2 4th OCTOBER 1988
30 DIM A% &270C
40 OSCLI"SAVE Debugged_D 38486A0
                                     +270C"
50 OSCLI"LOAD Debugged_D "+STR$~A%
60 FOR F=0 TO 3 STEP 3
70   P% = A% + &2230
80   [
90   OPT F
100  STMDB R13!, {R0-R9, R14}
110  .LOOP
120  LDRB R0, [R9], #1
130  CMP R0, #&7F
140  MOVHS R0, #&2E
150  CMP R0, #&1F
160  MOVLS R0, #&2E
170  SWI "XOS_WriteC"
180  BVS (LOOP+&35D)
190  SUBS R2, R2, #1
200  BNE LOOP
210  ]
220 NEXT
230 OSCLI"SAVE Debugged_D "+STR$~A%+
                                     " +270C"
240 OSCLI"SETTYPE Debugged_D &FFA"
```

- **EDIT's disappearing cursor** – If you use 'options' (<shift-f3>), you can change to a non-flashing cursor which is visible all the time, and you don't have to change it each time you switch on the

machine because EDIT stores its options in cmos ram. (Anyone know if there's a way to re-program it to change the colour of the cursor?)

- **Putting sound through the monitor** – It is perfectly possible to put the sound output from the 3.5mm stereo jack socket on the Archimedes through the amplifiers and speaker(s) of both the standard Acorn colour monitor and also the Phillips CM8833. All you need is a piece of twin screened cable (though actually I used ordinary thin three core mains flex which is OK for such a short length) and a 3.5 mm stereo jack plug (RS Components 274-284, £1.19 for two). The SCART plug connections are that pin 2 is right audio, 4 is earth and 6 is left audio. The pins should be numbered, but if not, 2, 4 and 6 are the first three pins along the long edge of the plug starting from the rectangular end.

Three possible problems: firstly, you may find that the SCART plug has no pins at positions 2, 4 and 6 in which case you will have to buy a new plug (Tandy N° 15-7030, £1.49 each) and re-wire the other connections. Secondly, you may find that your cable doesn't fit through the space available for the cable entry in the SCART plug – in which case, some judicious hacking of the plastic is needed. Finally, if the SCART plug has got pins at 2, 4 and 6, you may not be able to get at them to solder onto them. The solution here is to push each of the three pins back out through the body of the plug, solder them then push them back in again. It sounds easy but it's not – there are little barbs the stop the pins being pushed out so you have to try to manipulate these and push the pins out at the same time – OK if you have three hands!

The result – If the kids want to play Orion at full volume, they can, but I can turn it back down to a sensible volume when I use the system.

- **Stereo Speakers** – I recently followed the suggestion in Archive 1.2, page 7 and purchased the amplified speakers (Model 40-1259E) from a local Tandy store. Initially I also purchased a couple of mains adaptors to power each unit but I found that this caused a considerable mains hum through both speakers. Upon returning to the Tandy store, I was

advised that hum was caused by the mains adaptors not being earthed and, rather than spending more money on earthed mains adaptors, I exchanged the original adaptors for re-chargeable batteries which seem to work fine – there are also fewer wires trailing around! Whilst the stereo effect is very good, there is a problem in that there is no volume control fitted to these speakers; the suggestion from one of the staff at Tandy was to wire in two of their “L-Pad Speaker Level Controls”. Have any other Archive readers tried this or come up with a different solution?

I have found that by amending the \*Configure SoundDefault middle parameter to a lower value than 7, the volume can be reduced, this does provide some measure of control of sound volume; these parameters can take a value of 0 to 7 with each unit corresponding to one eighth of the overall maximum volume (see PRM, page 537).

• **Monitor problems (with solutions!) – Fujitsu and MAG 14C** – The problem began on the MAG with a distortion of the top few lines of the display when in a multisync mode. The nature of this problem suggested that the monitor took several scan lines to lock onto the composite sync signal fed to it from the Archimedes. It was decided to feed it with separate vertical (frame) and horizontal (line) sync signals, which necessitated having to alter the preset links 10 and 11 within the Archimedes. At present this involves cutting a track (Lk10) and inserting a link (Lk11), since these links are not brought out on the normal Molex type links. Having done this modification you need a \*Configure Sync 0 to set ‘Vertical sync only’ instead of ‘Composite sync’ to the monitor, otherwise the display rolls (indicating lost frame sync). This then cured the screen distortion problem but because of the way Acorn have put Composite/Vertical sync on the green signal, it kills the green output. With much trepidation this problem was overcome by disabling Acorn's implementation of this signal by cutting pin 2 of IC 4. The result is a perfect picture.

The Fujitsu monitor problem was much simpler and cured far easier after the troubles with the MAG.

Although the picture quality was very good, there was an excess of green, even though R39 had been removed (see Archive 1.3 p8) It was originally

thought that some internal adjustment of the green gun was needed but, with the experience of MAG monitor (and having tried it on the modified sync signal from the green output) the link changes proved unnecessary as the monitor worked perfectly happily with composite sync. Again the result was a perfect picture.

Anyone wanting further information on this modification can either contact Reg Dalton on 0742 487992 (less technical/Evenings only) or Steve Bass (who did all the work) on 0742 708028 (Daytime) or 0742 886622 (Evenings).

• **Screen saving from the SCML Teletext adaptor** – R L Jefferies has worked out how to dump screens from the SCML Teletext Adaptor. (Pressing the ‘S’ command option gives an error but SCML have not replied to his questions about that yet.) He has utilised Robin Newman's mode 7 to mode 9 utility as supplied on program disc 1.10. He copied this into the library directory on the Teletext disc and then created a mode 7 to mode 9 screensave module using Neil Strong's “Printkey screen dump” (issues 1.10 p 45 and 1.11 p 7). Two lines need altering in the source program:

line 840 should be changed to `cmp r0,#83` and line 920 should be `equs “M7M9 screensave scr<file>”`

Note that you should \*SETEVAL file XX before you load the “S” key module and before executing the Teletext module. One small niggle is that Robin Newman's module returns with the cursor ON. Is there an easy way to get rid of this?

• **Bug in C** – Karl Strickland writes... There appears to be a bug in the `fgets()` function in version 1.54A (latest public release) of ANSI C. Instead, use the function given below and declare it using

```
char *xfgets (char*,int,FILE*);
```

We have to use `xfgets` (as opposed to `fgets`) because the linker gets confused if it finds the same function declared twice.

```
char* xfgets (char* a,int b,FILE *s)
{
    int c,d;
    *a=NULL;          /* Must include
STDIO.H */
    for (c=0;c<b;c++) {
        d=fgetc(s);
```



```

if (d==EOF)
    return NULL;
else
    if (d==13)
        return a;
    else
        strncat (a, (char*)&d, 1);
}
return a;
}

```

• **A hint for Twin from Bjørn Fløtten** – When using Twin to edit large BASIC programs you can easily run out of memory. (Twin complains by hanging up the computer when called.) This is because of the inconvenient way Twin is called from BASIC. Because Twin works on ASCII files, BASIC has to expand the current program from tokenised form to ASCII. This copy is placed on top of the BASIC program and is maybe 20-30% larger than the original. Then Twin is called and makes a new copy of the program which it places on top of itself. This means that an unnecessary lot of memory is used.

One solution is to let Twin run from the address of PAGE. This works because Twin is not loaded into memory before the BASIC program is expanded. However, you will have to ensure that the BASIC program is bigger than Twin, if not, Twin will be loaded over the expanded version of your program.

(This unfortunately makes the problem of Twin not releasing some vectors worse. This means that these vectors will point into the middle of your BASIC program after exiting from Twin. Use reset after exiting to BASIC and then OLD, to cure this.)

• **Problems with (Master) EDIT** – If you get rubbish when you load up the EDIT ROM image from the BBC Master into the 6502 emulator, try loading it with:

```

*65arthur
*GO F800
*EDIT

```

• **From Arthur to BASIC** – If you are in the Arthur supervisor and you type \*Progame where Progame is a BASIC program, it will run the program and drop you back out into the supervisor (equivalent to \*BASIC -chain Progame).

However, if you \*LOAD Progame, it switches to BASIC, loads the program and gives you back control with a '>' prompt – equivalent to \*BASIC -load Progame but shorter to type and certainly easier than \*BASIC followed by LOAD "Progame". If you do a \*SHOW, you will see the filetype definitions which make sense of it all.

• **Sprites and Sprite Commands** – Paul Whitehorn writes... The Sprite Editor on the Welcome disc does not work properly being unable to create masked sprites (those with a transparent background) or to save shades of colour that have been set up in the various edit modes. For many board style games this is not an insuperable difficulty and for chess and checkers, mode 1 creates reasonably detailed sprites in red, yellow, black and white.

The simplest way of getting your own sprites from the Welcome disc onto your game or program disc is to \*SNEW to wipe out anything that can be called a sprite in RAM, \*SLOAD (by name or number) your sprite or sprite file into the machine from the Welcome disc (you don't need to be in the Sprite Editor to do this), take out the Welcome disc, insert your game disc, \*MOUNT it and \*SSAVE (name) the contents of the machine's RAM. Whatever sprites you have in RAM at this point will all go into a file created and named by the \*SSAVE command and will be placed on your own disc.

To check that they are all safely installed on your game disc \*SNEW again (to clear RAM) and \*SLOAD (name) the sprite(s) you have just \*SSAVE'd on your disc. Now typing \*SLIST will bring up a list of ALL the sprites in RAM which should correspond to the ones you have just transferred from the Welcome disc. At this point you can rename the sprites you see listed before you but the syntax for the \*SRENAME command is not as it appears in the User Guide. Type \*SRENAME (oldname) (newname) without commas or other punctuation marks separating the two names and without any quote marks round the names. This renames sprites in RAM and if you want these new names preserved on your disc then they must be \*SSAVE'd collectively back to disc and you can save them with a different file name if you wish.

\*SDELETE (name) deletes a sprite in RAM whereas \*DELETE (name) deletes a sprite (or file



# DRAWING BOARD

Drawing Board is a drawing package that outputs to most dot matrix and colour dot matrix printers.

With DB you can draw anything - from Xmas cards to finished PCB layouts - print it out at 200 dots/inch in colour or black and white (depending on your printer).

Complex drawings can be created on several 'planes' (like layers of see-through paper) then displayed as a single picture; or planes can be printed separately, perfectly registered.

This advert is a single DB file (originally printed out A4).

Drawing Board does not run on the 305.

## £59

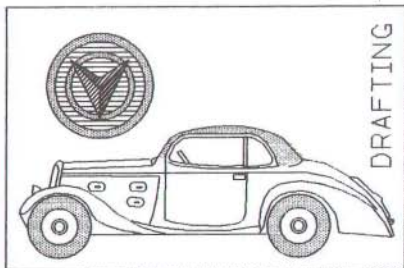
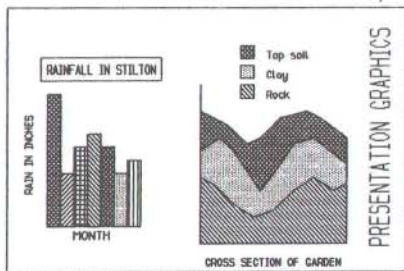
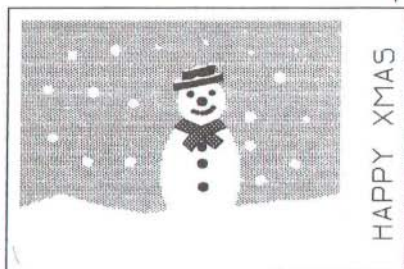
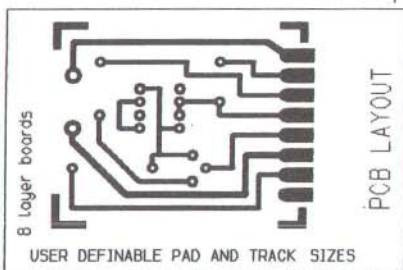
INC VAT + P&P

Please make cheques payable to JANSONS

# JANSONS

93 NORTH STREET • STILTON  
PETERBOROUGH • PE7 3RR

TEL (0733) 244702



of sprites) on disc. Having weeded out sprites you don't require with \*SDELETE it is best to \*SSAVE those you do need (possibly with a new file name) and \*SNEW to clear RAM before proceeding further. This way you will not have rogue sprites lurking in RAM or in files on your disc where they may be called by default if they have the same name or number as the new ones.

To create files of sprites from different sources you can use \*SMERGE which merges a sprite (or file of same) from disc with those already installed in RAM. Those coming from disc have priority so if you have two sprites with the same name or number, the one on the disc will overpower its namesake in RAM. This command seems a little haphazard on my 310 especially with sprites which are numbered and occasionally several tries are needed to make it happen as required.

If you intend to choose sprites during the course of your game as a consequence of numerical operations then it is necessary to use numbers as names so that they can be called with the VDU command. The VDU call takes variables with a range between 0 and 255 so make sure your sprites don't exceed this range.

\*SINFO prints the size of the sprite workspace and \*SCOPY (oldname) (newname) copies the named sprite. \*SGET picks up a portion of the screen as a sprite but for details of this see the User Guide.

• **EMR SoundSynth + CC ROM podule** – There has been a problem of a clash between EMR's SoundSynth and Computer Concepts' ROM/RAM podule in that after using SoundSynth, the contents of the RAM filing system disappeared! (Fortunately, I had a recent backup!) EMR have been looking into the problem and have sorted it out. If you have had the same problem, return your disc to EMR and they will replace it "in most cases, free of charge".

• **Loading palette files** – Here is a simple BASIC program to load a "!Palette" file and set the palette up. The file format really is simple: 3 bytes are stored for each colour – red, green and blue respectively. This group of 3 bytes is repeated for all 20 colours saved – logical colours 0-15, the border, and mouse colours 1-3.

```
10 REM >TestPal
20 REM ==> To demonstrate reading a
30 REM ==> Desktop !Palette file
40 REM By John Smith, September 1988
50 REM
60 MODE 12 :REM Any 16-colour mode
70 : will do.
80 REM Put all 16 colours on screen,
   and a 2-colour mouse pointer.
90 FOR x%=0 TO 15
100   GCOL x%
110   RECTANGLE FILL x%*80,0,80,1023
120 NEXT
130 MOUSE ON
140 *POINTER 1
150 PRINTTAB(0,0);"Now press a key to
   load the '!Palette' file"
160 dummy=GET
170 :
180 REM Now load the palette file.
190 handle%=OPENIN "!Palette"
200 REM Do "standard" colours first.
210 FOR loop%=0 TO 15
220   COLOUR loop%, BGET#handle%,
       BGET#handle%, BGET#handle%
230 NEXT
240 :
250 REM Screen Border
260 VDU 19,0,24, BGET#handle%,
       BGET#handle%, BGET#handle%
270 :
280 REM Mouse colour 1
290 MOUSE COLOUR 1, BGET#handle%,
       BGET#handle%, BGET#handle%
300 MOUSE COLOUR 2, BGET#handle%,
       BGET#handle%, BGET#handle%
310 MOUSE COLOUR 3, BGET#handle%,
       BGET#handle%, BGET#handle%
320 CLOSE#handle%
```

• **InterWord to ArcWriter** – if you have InterWord files from BBC days and want to transfer them to ArcWriter, look in the First Word Plus hints below where the transfer to FWP is achieved by transferring first to ArcWriter.

• **Double-precision to BBC BASIC** – following on from the floating point BCD to BBC BASIC conversion program published in Archive 1.11, page 32, Carl Cepurneek has done a routine which converts from the more efficient Double-precision reals to BBC BASIC.

```
10 REM>Carl - FPU number conversion
20 REM BBC BASIC V to FPU memory
30 REM format conversions
```



```

40 REM Packed decimal print routine
50
60 REM Carl P.Cepurneek 26 Clyde St
70 REM Parkside, S.A.5063,Australia
100
110 DEFPROCcvt_s(bbc,adr) :REM from
    BBC real at |bbc to FP at adr
120 LOCAL exp%,sign%,mant%
130 !adr=0:adr!4=0 :REM clear
140 IF |bbc=0:!adr=0:ENDPROC
150 exp%=(bbc?4)-&80 :REM get
    adjusted exponent
160 IFexp%>0 exp%=exp%+126 ELSE exp%=
    126-ABS(exp%)
170 !adr=!adr OR exp%<<23:REM set
    exponent bits
180 sign%=!bbc AND &80000000
    :REM isolate sign bit
190 !adr=!adr OR sign%:REM set sign bit
200 mant%=!bbc AND &7FFFFFFF :REM
    clear neg bit if set
210 !adr=!adr OR mant%>>>8:REM set
    mantissa
220 ENDPROC
230
240 DEFPROCcvt_d(bbc,adr):REM from
    real at address bbc
250 LOCAL exp%,sign%,mant%
260 !adr=0:adr!4=0
270 IF |bbc=0:!adr=0:ENDPROC
280 exp%=(bbc?4)-&80 :REM get
    adjusted bbc exponent
290 IF exp%>0 exp%=exp%+1022 ELSE exp%
    =1022-ABS(exp%)
300 !adr=!adr OR exp%<<20
310 sign%=!bbc AND &80000000
    :REM isolate sign bit
320 !adr=!adr OR sign%:REM set sign bit
330 mant%=!bbc AND &7FFFFFFF
    :REM clear sign bit set
340 !adr=!adr OR mant%>>>11 :REM set
    mantissa msbits
350 mant%=mant%<<21 :REM shift over
    low bits
360 adr!4=adr!4 OR mant% :REM set
    mantissa lsbits
370 ENDPROC
380
390 DEFPROCcvt_e(bbc,adr):REM from
    BBC real at |bbc
400 LOCAL exp%,sign%,mant%
410 !adr=0:adr!4=0:adr!8=0 :REM clear
    destination
420 IF |bbc=0:!adr=0:ENDPROC
430 exp%=(bbc?4)-&80 :REM get
    adjusted exponent
440 IFexp%>0 exp%=exp%+16382 ELSE
    exp%=16382-ABS(exp%)
450 !adr=!adr OR exp% :REM set
    exponent bits
460 sign%=!bbc AND &80000000
    :REM isolate sign bit
470 !adr=!adr OR sign%:REM set sign bit
480 mant%=!bbc OR &80000000 :REM set
    top bit (J)
490 adr!4=adr!4 OR mant% :REM set
    mantissa
500 ENDPROC
510
520 DEFFNprint_packed(base%)
530 LOCAL ms$,ms$,es$,es$,m$,s%
540 @%=&00001
550 m%=(!base% AND &80000000)>>>31
560 IF m%=0 ms$="+" ELSE ms$="-"
570 PRINT ms$;
580 m%=(!base% AND &F<<8) >>>8
590 PRINT m%;".";
600 FOR s%=4 TO 0 STEP -4
610 m%=(!base% AND &F<<s%) >>>s%
620 PRINT m%;
630 NEXT
640 FOR w%=4 TO 8 STEP 4
650 FOR s%=28 TO 0 STEP -4
660 m%=(w%!base% AND &F<<s%) >>>s%
670 PRINT m%;
680 NEXT
690 NEXT
700 e%=(!base% AND &40000000)>>>30
710 IF e%=0 es$="+" ELSE es$="-"
720 PRINT "E";es$;
730 FOR s%=24 TO 12 STEP -4
740 m%=(!base% AND &F<<s%) >>>s%
750 PRINT m%;
760 NEXT
770 =" "

```

• Using 40-track drives – If you have a 40 track 5.25" drive as used on PC clones, you can use it under the PC emulator but, in native Archimedes ADFS modes you will have problems. To get round it, format the disc as normal then, when it goes beyond "Formatting 40" and starts clicking as it reaches the end of the travel on the head movement, press <ctrl-break>. The disc is now formatted, but the free space map and catalogue have not been installed. This can be done with:

```

DIM buffer% 3072
SYS"ADFS_DiscOp",,1,0,buffer%,3072
SYS"ADFS_DiscOp",,2,1,buffer%,3072

```

*You try this entirely at your own risk! Ed.*



• **Problems with ON ERROR LOCAL** – John Smith says he thinks he's discovered a problem when using LOCAL ERROR with ON ERROR LOCAL. It occurs because ON ERROR LOCAL stores the exact position in the program structure. The User Guide mentions this with respect to REPEAT...UNTIL and WHILE...ENDWHILE loops, but it also seems to apply to IF..THEN...ELSE...ENDIF constructs. Once the error handler has done its job, processing must continue at the same part in the structure in which the original error happened. For example,

```
1000 DEF PROCtest(arg)
1010 LOCAL ERROR
1020 ON ERROR LOCAL PRINT "Can't
      divide by zero - try again"
1030 IF arg < 100 THEN
1040 INPUT value
1050 PRINT arg/value
1060 ENDIF
1070 ENDPROC
```

If a value of zero is input, line 1050 will error. As written, the above code will then print a warning message and effectively execute a "GOTO 1030". At this point, the error handler stack is corrupted. The program may appear to work without problem, but if any future error should occur, the program may produce a run-time error something like "Attempt to exit from badly nested error handler" or something similar. Occasionally, I have even managed to produce the dreaded "Address Exception error"!

The solution to this problem is quite simple – place the code that could error together with the relevant error handler in a separate PROCedure. I have not (yet) had any problems doing this:

```
1000 DEF PROCtest(arg)
1030 IF arg < 100 THEN
1050 PROCprint(arg)
1060 ENDIF
1070 ENDPROC
2000 DEF PROCprint(arg)
2010 LOCAL ERROR
2020 ON ERROR LOCAL PRINT "Can't
      divide by zero - try again"
2040 INPUT value
2050 PRINT arg/value
2060 ENDPROC
```

## First Word Plus Hints & Tips

Here are a few of the many contributions we've been sent about First Word Plus. We've got a file of about ten other contributions, so we do need someone who knows FWP, has a reasonable skill at writing and has a bit of spare time to field the enquiries, assimilate the hints and tips and regurgitate something to us each month.

• **First Word Plus startup** – If you are getting fed up of starting First Word Plus and finding yourself in lower case, you could \*Configure Nocats, but that is a nuisance if you do programming and want to start up in Shiftcaps. The solution is to extend the boot file, adding \*FX202,48.

• **LQ850 with sheetfeeder** – If you're having problems with this combination, Acorn say that you should edit the printer driver so that entry 4 (vertical tab) is commented out by pre-fixing it with a "\*". Also, note that if your printer is doing auto-linefeeds, you should also edit option 1 to remove the line feed value (A) from the line to leave just the carriage return (D).

• **Star LC10** – (ref. the Help enquiry we had last month) David Francis says he uses the Epson FX driver successfully with this printer. He has the dip switches set as follows: 1-1 to 1-8 and 2-1 and 2-4 are all ON and 2-2 and 2-3 are OFF. The Archimedes is configured to IGNORE 0. With this setting, the printer prints out the test, including the graphics part. As set, it also prints under <ctrl-B> as well as with Artisan. He's also tried to change the hex file to get it to work with quadruple size, but hasn't succeeded yet.

• **Printer drivers** – In designing my own printer drivers for both the Citizen 120D and Star LC-10 printers, I have not come across the problem noted, but mine were modified from the Epson LX printer driver. There was an item on the Letters section (page 128) of October's issue of the Micro User referring to an apparently similar problem, which the writer cured by "commenting out" line &20 of the printer driver. I must admit that I cannot understand why this should have any effect, but in case it benefits other Archive readers, I detail below those elements of the printer drivers I have set up relating to form or line feeds (refer also to page 227 of the First Word Plus manual):

1D, A – Linefeed (N.B. carriage return & linefeed)  
 1E, C – Formfeed (N.B. &C = 12 as mentioned in the article)

1F, 12 – Horizontal initialisation (N.B. appears to cancel condensed)

20, 1B, 52, 0 – Vertical initialisation (N.B. appears to select USA characters)

A further possible cause of problems may be to do with the printer's dip switches, on page 218 of the First Word Plus manual it does state that:

Skip over perforation must be DISABLED and Auto linefeed must be DISABLED

• **Printer Drivers with extended characters** – Ted Peat has sent in various bits for FWP which we've put on the program disc.

The printer drivers were written to permit both the Panasonic KXP-1081 and the Diablo-ecs printer to access the same set of extended characters (those with codes from 160 to 255). The character set of the Archimedes is also modified to suit.

The Diablo-ecs printers use a daisy wheel with a double row of characters, giving 192 in all. The usual ASCII set is supplemented by characters with codes >128. The description given here applies to the 'scientific' wheel containing Greek and mathematical symbols. The KXP-1081 can be switched to emulate the IBM Graphics set, which also includes some Greek letters and some (but not all) useful mathematical symbols, again with codes >128. There are also some table-drawing symbols which are replicated several times over, and accented characters not needed for this application. Since up to 40 user-defined characters are possible, these redundant characters can be replaced. This then offers the opportunity to produce matched sets of symbols for the two printers and the Archimedes.

I decided to include the complete Greek alphabet, which involves duplication of some upper case letters, and (for the KXP-1081 only), one set of table-drawing symbols. Two other symbols were added: an overline and backspace. Backspace is entered into the text as a large leftward pointing arrow-head (stored as character 161), but interpreted by the printer as a true backspace. One consequence of this, of course, is that text loses its WYSIWYG property. The advantage, however, is

that both superscripts and subscripts can be attached to the same symbol, and the overline can be used to write a bar above a character. Since these peculiar needs are only likely within equations, the loss of on-screen text formatting is a minor irritation. Another complication was that the Diablo lost its left margin alignment after a backspace, when printing in the reverse direction. This option had to be switched off, slowing the printer down.

There were two options for changing the Archimedes character set. One was to change the Archimedes font, to match that desired as far as possible, and then to redefine characters for the few changes needed. The most useful character set is that described as BFONT on p. 458 of the User Guide. To access this you need to \*Configure Country Master. The second way: which is the one I adopted was to redefine all characters from 160 to 255. To do this I added a section to the 1st Word + loading program (Library.1stWord+ on the program disc).

Once this was done, mapping the new character set to the Diablo's set was easy: merely rewrite one of the provided Diablo Printer Drivers (in the hex directory) by changing the character translation table which forms the last part of the printer driver text.

The KXP-1081 offered more complications. Firstly, it is necessary to download the redefined character set before loading 1stWordPlus. Another addition to the loader program, together with a user prompt, took care of this. Secondly, switching this printer to its IBM Graphics mode caused complications with graphics printing: all the linefeeds became doubled! The printer driver codes were therefore expanded so that for each character the printer was switched into IBM Graphics set, the character printed, and then switched back to the Epson mode. This sends seven characters to the printer for every one actually printed. Since these are only occasional characters, the loss of efficiency is not important

• **Printer Driver Generator** – John Smith has sent in, not another printer driver, but a printer driver generator program. He's given it to be used as public domain software, so we have put it on Shareware disc number 2. For best results, copy the PDG program onto a copy of the 1st Word Plus Utilities disk before running it.



• **Tabs and underlining** – This one is for dummies like me, writes David Crofts, who learn word-processing at the finger tip.

**Underlining** – I spent many happy, but frustrating, hours fiddling with forms to try to sort out underlining in intervening spaces, till I discovered that the hated TAB holds the key! (I expect this is obvious for lots of people, but hold on to the yawns.) I was used to VIEW where the TAB key worked “properly”. (It inserted a TAB character, whatever that was, but it meant that a change in ruler meant a change in layout.) In FWP the Indent key does this job, except that it puts in a special stretch space which is fixed in spite of further ruler alterations. Because of this I had ignored TAB as useless.

TAB inserts a set of normal spaces which have to be deleted individually – a nuisance if you are experimenting with documents. But – you can underline TAB spaces, whereas you can't underline Indents! And therein lies the hint!

By all means use indents to facilitate experimentation but, in the end, TABs are preferable, where you wish subsequently to underline. If you have used Indents, then all is not lost, though fiddle is the name of the game. Turn off Insert, and turn on Underline (f2), then fill in the spaces with Fixed Space (f10). Then use the mouse or keyboard to underline the words.

e.g. Say you have a form, the head of which is:

<u>Name</u>	<u>Address</u>	<u>Telephone</u>
-------------	----------------	------------------

the above is the result using Indent, but using TAB you will achieve:

<u>Name</u>	<u>Address</u>	<u>Telephone</u>
-------------	----------------	------------------

Placing a TAB at the appropriate point near the end of the ruler allows the line to be printed to the same length on each line.

• **InterWord to FWP via ArcWriter** – This is how Ian Barnes achieves the transfer:

1. Transfer file from IW to AW\_DOC directory
2. Run ArcWriter
3. Press <select> for filing menu
4. Highlight file x
5. Press <menu>, select import, flowing, return
6. Press <select> for filing menu

7. Press <menu> and select export plus new name
8. Transfer file to 1WP.doc directory
9. Run 1WP and open file
10. Select WPmode from edit menu
11. Tidy up text by deleting control characters, changing any odd letters and inserting returns at line ends to leave a presentable result

• **Files from ArcWriter** – The article by Brian Carroll in July's issue of Archive under the heading of “First Word Plus Notes” (pages 21-23) was very useful together with the program ConvertWP (also on that month's Magazine disc). Although designed to convert Wordwise Plus files, the program works quite well with ArcWriter as well. Briefly the procedure to convert ArcWriter files into First Word Plus is as follows:

1. From the 'Filing' menu in ArcWriter select the 'Export' option to save the file (e.g. under the name “testARC”) and then exit from ArcWriter.
2. Load the ConvertWP program and enter the appropriate Source (“testARC”) and Destination (e.g. “test1WP”) filenames.
3. Load First Word Plus and then the converted file (“test1WP”).
4. Select 'WPmode on' from the Edit menu.
5. Reformat the whole text by using the Style menu to 'Reformat' the 'Whole document'.

Odd words may sometimes get missed out of the converted text, possibly due to the fact that this data was preceded by “tabs” in ArcWriter. **A**

## Small Ad's

**Radioactive Nuclides program.** Shows decay chains, half lives, abundances and decay modes of 2000+ isotopes. Instructive. Useful for colleges. Requires 1 Mbyte + M/S monitor. £5.95. R.W.Darlington, 1 Fells Grove, Worsley, M28 5JN.

**Parlour games:** 3's and 5's (dominoes), strip jack naked, concentration. £3.95 to R.W.Darlington, 1 Fells Grove, Worsley, Manchester, M28 5JN.

**For Sale:** Euclid £25, Hoverbod £5, Terramex £10. Please phone Chesterfield (0246) 270730 after 6.

**For Sale:** System Delta Plus (almost new) £35, Acorn ROM podule (new) £35. (0536) 724981. **A**



# **EXPANSION CARDS FOR THE ACORN ARCHIMEDES COMPUTER SYSTEM**

**IEEE488 INTERFACE** a full implementation of the standard for automatic test and measurement systems

**16 BIT PARALLEL I/O** two 16 bit input or output ports with handshake lines for digital control applications

**DUAL RS423 SERIAL INTERFACE** for communicating with two additional RS423 or RS232 devices eg printers, plotters, instruments, etc

**12 BIT ANALOGUE I/O** in development

All the above high performance expansion cards are supplied with high level software for ease of use and a comprehensive user guide.

Take advantage of Intelligent Interfaces' expertise and purchase a complete Archimedes Computer System.

Officially appointed Acorn Scientific Dealer.



**Intelligent Interfaces Ltd**

43b Wood Street  
Stratford-upon-Avon  
Warwickshire  
CV37 6JQ

Tel: 0789 415875

Telex: 312242 MIDTLX G



Now available

£105 from Archive

## PipeDream

PipeDream is now available on the Acorn Archimedes. It provides comprehensive word processing, spreadsheet and database facilities integrated in a way only dreamed of before.

With other integrated packages, you have to divide your work into artificial sections, such as text, numbers and calculation, and database.

With PipeDream, you compose your work in the order you want to print it, with text and numbers all together in one document. Incorporate calculations directly into paragraphs of text and formatted paragraphs directly into spreadsheets.

PipeDream is ideal for all tasks involving words and numbers. From writing thank-you letters to encyclopaediae, invoices to cash flow forecasts, stamp-collection records to inventory management, film scripts to mail-shots.

PipeDream is 100% file and keystroke compatible with Z88 PipeDream and PipeDream on the IBM PC. It is also compatible with View Professional on the BBC microcomputer. You can create documents on any of these computers, transfer the files to any other and continue working, with no loss of information. No other software enables you to share your files with all these computers.

PipeDream for the Acorn Archimedes costs £99 + VAT.

It is not possible to detail all of PipeDream's facilities here. For full details or to order PipeDream call us on 0954 211472 or write to us at Colton Software, Broadway House, 149-151 St Neots Road, Cambridge CB3 7QJ.

PipeDream - power at your fingertips.



# Fractal Landscapes

## Malcolm Banthorpe

The program generates quite convincing random landscapes (actually a number of islands in a seascape) in the following manner:

The starting point is a horizontal rectangular plane. The coordinates of its centre point are calculated in order to divide it into four equal smaller rectangles. This centre point is displaced by a random distance, at right-angles to the plane of the original rectangle. The process is then applied to the four smaller rectangles and is recursively applied to the successively smaller facets formed until a suitable size for plotting is reached. The facets thus calculated are not strictly speaking rectangles as their four vertices do not lie in the same plane and the corners are not right-angles, but it's probably easier to think of them as such. The result is that the original rectangular plane becomes folded in a random manner which to some extent simulates the folding of the Earth's crust.

Using standard 3D techniques, any facets which now lie above a predetermined height, representing sea level, are plotted. Each is assigned a colour according to its height above sea level and is shaded to enhance the 3D effect. The colours are selected to represent sand, foliage, rock etc.

The size of the facets plotted is determined in the BASIC program by the variable, "resolution%" and this may be set to 4 to show more detail - at the expense of increasing the execution time considerably. Each time the figure is halved, the plotting time will be increased by a factor of four. The variable may also be set to a larger value, say, 64 in order to make the individual facets more visible and to perhaps clarify how the program works.

Since the program generates a different random landscape every time, not every one will be a winner; sometimes most of the "landmass" generated will be below sea level and only a few small islands will be visible. In this case just run the program again for another try. Execution time also depends partly on how much land is visible. This fortunately means that the less successful attempts take less time to plot.

The first listing is in C and the second is in BASIC. If you haven't got a copy of C, there is an object code version on the monthly program disc.

```
/* fractal landscape generator
   by Malcolm Banthorpe */

#include <Arthur.h>
#include <Stdio.h>
#include <Stdlib.h>
#include <Signal.h>

#define OS_Plot 0x45

int main(void)
{
    void fold(int, int, int, int, int,
              int, int, int);

    void fn(int);
    reg_set s;
    int a, d;
    signal(4, *fn);
    mode(15);
    gcol(0, 32);
    tint(2, 2);
    rectanglefill(0, 0, 1279, 800);
    gcol(0, 48);
    tint(2, 3);
    rectanglefill(0, 800, 1279, 223);
    swi(OS_ReadMonotonicTime, &s);
    for (d = 0; d <= s.r[0] % 123456; d++)
        a = rnd(1234);
    vduq(29, 128, 2, 0, 2);
    fold(-640, 800, 640, -400, 0, 0,
        0, 0);
}

int sqr(n)
int n;
{
    int c = 0;
    int ng = n>>1;
    int g;
    if (n <= 1)
        g = 0;
    else
        do
            ng = (n/(g = ng) + g)>>1;
            while ((10 >= c++) && (g != ng));
    return (g);
}
```

```

void plot3D(k, x, y, z)
int k, x, y, z;
{
    reg_set s;
    z += 1000;
    s.r[0] = k;
    s.r[1] = (x << 10) / z;
    s.r[2] = (y << 10) / z;
    swi(OS_Plot, &s);
}

void fold(x1, y1, x2, y2, z1, z2, z3,
          z4)
int x1, y1, x2, y2, z1, z2, z3, z4;
{
    int x3, y3, z5, v, r, g, b, l, x,
        y, z, h;

    if ((y1-y2) < 4)
    {
        if (z2 < -4)
        {
            r = 2; g = 2; b = 1;
            if (z2 < -8 - rnd(8) - rnd(8))
            {
                r = 0; b = 0; g = 1;
                if (z2 < -32 - rnd(40) -
                    rnd(40))
                {
                    r = 1;
                    if (z2 < -50 - rnd(40) -
                        rnd(40))
                    {
                        b = 1;
                        if (z2 < -110 - rnd
                            (40) - rnd(40))
                        {
                            r = 2; g = 2; b
                                = 2;
                        }
                    }
                }
            }
        }
        x = x1 - x2;
        z = z1 - z2;
        v = 4 + (z<<2)/sqr(x*x+z*z);
        if (v == 8) v = 7;
        l = v >> 2;
        gcol(0, r+l + ((g+l)<<2) +
            ((b+l)<<4));
        tint(2, v % 4);
        y = (y1 * 3547 + (x1 << 11))
            >> 12;

```

```

        plot3D(4, (x1*3547-(y1<<11))
            >>12,
            ((y<<11)-z1*3547)
            >>12,
            ((z1<<11)+y*3547)
            >>12);
        y = (y1*3547+(x2<<11))>>12;
        plot3D(4, (x2*3547-(y1<<11))
            >>12,
            ((y<<11)-z2*3547)
            >>12,
            ((z2<<11)+y*3547)
            >> 12);
        y = (y2*3547+(x2<<11))>>12;
        plot3D(117, (x2*3547-(y2<<
            11))>>12,
            ((y<<11)-z3*
            3547) >> 12,
            ((z3<<11)+y*
            3547)>>12);
    }
}
else
{
    h = (x2-x1)>>1;
    x3 = (x1+x2)>>1;
    y3 = (y1+y2)>>1;
    z5 = ((z1+z2+z3+z4)>>2)+rnd(h)
        +rnd(h)-h;
    fold(x3,y1,x2,y3,(z1+z2)>>1,z2,
        (z2+z3)>>1,z5);
    fold(x1,y1,x3,y3,z1,(z1+z2)>>1,
        z5,(z4+z1)>>1);
    fold(x3,y3,x2,y2,z5,(z2+z3)>>1,
        z3,(z3+z4)>>1);
    fold(x1,y3,x3,y2,(z4+z1)>>1,z5,
        (z3+z4)>>1,z4);
}

void fn(type)
int type;
{
    int t;
    switch (type)
    {
        case SIGINT :
            printf("Escape\n");
            printf("type G0 to re-start
                \n");
            exit(0);
            break;

```



```

default :
  printf("the signal type was
          %d\n", type);
  t = raise(SIGILL);
}

10 REM >landscape
20 REM by Malcolm Banthorpe
30
40 A%=RND(-TIME)
50 sea_level% = -5
60 resolution% = 16
70 MODE15
80 OFF
90 GCOL 48 TINT 192
100 RECTANGLE FILL 0,1023,1279,-300
110 GCOL 32 TINT 0
120 RECTANGLE FILL 0,800,1279,-923
130 ORIGIN 640, 512
140 cos55 = COSRAD55
150 sin55 = SINRAD55
160 cos30 = COSRAD-30
170 sin30 = SINRAD-30
180 PROCfold(-640,600,640,-512,0,0,0,
              0)
190 END
200
210 DEFPROCfold(x1%,y1%,x2%,y2%,z1%,
               z2%,z3%,z4%)
220 LOCAL x3%,y3%,z12%,z23%,z34%,
          z41%,z5%
230 IF (y1%-y2%) < resolution% THEN
240   IF z1% < sea_level% THEN
250     R%=2: G%=2: B%=1
260     IF z1% < -15 -RND(40) THEN
270       R%=0: B%=0: G%=1
280       IF z1% < -32 -RND(40) THEN
290         R%=1
300         IF z1% < -64-RND(40) THEN
310           B%=1
320           IF z1% < -110 -RND(32)
              R%=2:G%=2:B%=2
330         ENDIF
340       ENDIF
350     ENDIF
360     V%= 4*(1+(z1%-z2%)/SQR((x1%-
              x2%)*(x1%-x2%)+ (z1%
              -z2%)*(z1%-z2%)))

370     L% = V%>>2
380     T% = V%MOD4

390     GCOL R%+L%+( (G%+L%)<<2)+
              ((B%+L%)<<4) TINT T%<<6
400     y% = y1%*cos30 - x1%*sin30
410     X% = x1%*cos30 + y1%*sin30
420     Y% = y%*cos55 - z1%*sin55
430     Z% = z1%*cos55 + y%*sin55
440     PROCplot3D(4, X%, Y%, Z%)
450     y% = y1%*cos30 - x2%*sin30
460     X% = x2%*cos30 + y1%*sin30
470     Y% = y%*cos55 - z2%*sin55
480     Z% = z2%*cos55 + y%*sin55
490     PROCplot3D(4, X%, Y%, Z%)
500     y% = y2%*cos30 - x1%*sin30
510     X% = x1%*cos30 + y2%*sin30
520     Y% = y%*cos55 - z4%*sin55
530     Z% = z4%*cos55 + y%*sin55
540     PROCplot3D(85, X%, Y%, Z%)
550     y% = y2%*cos30 - x2%*sin30
560     X% = x2%*cos30 + y2%*sin30
570     Y% = y%*cos55 - z3%*sin55
580     Z% = z3%*cos55 + y%*sin55
590     PROCplot3D(85, X%, Y%, Z%)
600   ENDIF
610   ELSE
620     h% = (x2% - x1%)>>1
630     x3% = (x1% + x2%)>>1
640     y3% = (y1% + y2%)>>1
650     z12% = (z1% + z2%)>>1
660     z23% = (z2% + z3%)>>1
670     z34% = (z3% + z4%)>>1
680     z41% = (z1% + z4%)>>1
690     z5% = ((z1%+z2%+z3%+z4%)>>2)
              +RND(h%)+RND(h%)-h%
700     PROCfold(x3%,y1%,x2%,y3%,z12%,
               z2%,z23%,z5%)
710     PROCfold(x1%,y1%,x3%,y3%,z1%,
               z12%,z5%,z41%)
720     PROCfold(x3%,y3%,x2%,y2%,z5%,
               z23%,z3% z34%)
730     PROCfold(x1%,y3%,x3%,y2%,z41%,
               z5%, z34%, z4%)
740   ENDIF
750 ENDPROC
760
770 DEFPROCplot3D(k%, x%, y%, z%)
780 PLOT k%, (x%<<10) DIV (1000 + z%)
              , (y%<<10) DIV (1000+z%)
790 ENDPROC A

```

# Help Needed and Offered

• **Chess** – Is there a chess program for the Archimedes? or even a BBC one that works under the emulator? (P.Snook, Rotherham)

• **Econet help!** – Firstly, let's have some hints and tips and/or major articles about using Archimedes on Econet. Secondly is there an equivalent of COPYF or preferably the S.J. utility MULTICOPY to copy complete discs from the 310 to the net and vice versa? Thirdly, can MS-DOS files be pulled off the net in emulator mode as it can with the local floppy or hard disc? (John Rogers, St Joseph's School, Swindon.)

• **Compilers/Linkers on Econet** – It would be useful (on Econet especially) to be able to load a compiler and/or linker and submit different source files to it repeatedly without having to re-load the compiler. This would allow a teacher, say, to set up one machine for compiling Pascal (or C etc) and another for linking. This would save the tremendous network traffic associated with repeatedly loading such software. (Michael Ryan, Dundee)

• **Acorn 'Prestel Adaptor'** – can this be used on the Archimedes? If so, how? With Hearsay, for example? (W.D.Smith, Edinburgh)

• **First Word Plus** – Is there anyone out there who would be prepared to edit all the First Word Plus material we are getting? There's such a lot of it that it needs someone who is actually using the software to see which bits are most relevant and check that there is no duplication. Any offers?

• **First Word Plus** – Any way round the problems of having to change the auto-linefeed dip switch every time you want to put in graphics or use First Fonts? (Tom Johnson, Plymouth)

• **First Word Plus** – When trying to print a sprite in with some text in First Word Plus it came out with horizontal blank lines in between, then the printer fed the paper through for about 20 line before following with the next bit of text. (Gary Baum, Cumbria who has a Kaga 815, Epson compatible.)

*This just illustrates what I said about needing someone to take on as FWP editor – someone to whom I can send such queries (I don't use FWP seriously enough to be able to answer them) – and*

*that person could answer the correspondent directly and then explain it in the magazine for the benefit of others facing the same problem. Ed*

• **Keyboard cable extender** – Does anyone know of a supplier of a lead to extend the Archimedes keyboard cable? (Roger Morris, Warwick)

• **Machine code sorting** – Has anyone got a machine code routine for sorting 800 or so strings of variable length (up to ~50 characters)? (Nigel Kirby, Oldham)

• **Eizo 8060S multi-sync monitor** – How do you centre the display in hi-res modes? (Fred Lee, Singapore)

• **File corruption** – Is there any way to rescue a file when it says in the directory that the file length is 0? (Claus Birkner, W. Germany)

• **Elliptical circles on Screen dumps** – Is there any way of getting round (sic) the problem of printer dumps (e.g. Gerrald Fitton's Epson mode 7 dump) that turn ellipses into circles? (Gary Baum, Cumbria)

• **Programming Competitions** – Is there anyone who could set (and adjudicate) a programming competition? I'm sure we could find some small prize to act as an incentive for the competitors.

• **Wild-carding (\* and #) and case-insensitivity** – these are both exhibited by the ADFS filing system. Are these routines available as SWI's? or could someone write routine for us to use on our own strings? (Roger Darlinton, Manchester)

• **C programming** – Does anyone know what happens to a value returned from main() in a C program? Also, does anyone know how to use the re-locatable switch on the linker; the linker displays it as an option if you enter link -help; does it work at all on the Archimedes? Does anyone know how to build libraries of functions with the C compiler – using something like Microsoft's Librarian program for the PC? Is anyone feeling the need for a MAKE utility? These are supplied standard with many MSDOS compilers and with UNIX. Is there any demand for a utility of this sort? (Karl Strickland, Liverpool)



*(I think I remember someone telling me on the phone, that they had worked out how to create relocatable modules with C... but I can't remember who... Also, Alan Barclay, I think, says he has a program which will turn any program in BASIC, C, Pascal etc into a relocatable module – all we have to do is persuade him to release it on PD which I think he hinted he might – or possibly market it for a relatively small fee. If anyone is interested, write to Alan c/o the Archive office.)*

• **MicroLink modem** – Has anyone actually used a MicroLink Multi-speed modem on the Archimedes? Will the lead and the software supplied with the IBM-PC version work OK on the Archimedes? (Karl Strickland, Liverpool)

• **ArcTFS Review** – Did I send ArcTFS to someone for review? I've forgotten who it was and didn't write it down. Sorry!

### Help offered

• **Intelligent \*COPY** – We asked last month for a way of backing up a hard disc intelligently, i.e. those files with a particular date or after a particular date. Brian Carroll suggests that the ARC utility which comes as PD from Beebug and which we put on the issue 11 program disc, will do this.

• **Solidisk Teletext adaptor** – Mr R.L. Jefferies replied to Mr Carey's Help!! plea about downloading software by a letter sent through the Archive office. If anyone else wants the same info, send an S.A.E. and we'll pass it on to Mr Jefferies – I haven't had time to ask for permission to publish his address.

Also, Peter Creed of Dudley Micros on 0902-342214 is offering help on Solidisk TTX adaptor. Also, see Hints and Tips on page 7.

• **Lisp for the poor** – Anyone who finds Acorn's price for Lisp a bit steep ('only' £210 through Archive!) may now have a solution in the form of IBM PD/Shareware. There is a program around called "Marc Alder's Lisp Interpreter" which comes complete with C source code. The program supports a subset of Lisp and it is possible to re-compile it with ANSI C – it seems to work OK but I have no experience of Lisp. I'm not sure if it's Shareware or PD though, so I don't know about the legal position. Does anyone?

• **Reading non-standard discs** – There is an article in the November issue of Acorn User (page 46) together with a program listing. This deals with the transfer of IBM MS-DOS (and Atari ST TOS) format discs to Archimedes ADFS format without using the PC Emulator's Getfile and Putfile commands. There is only a simple version in Acorn User's program listing; the full version is apparently included on their November 3.5" Archimedes disc. I assume that this program could be suitably modified for other computers, by those with sufficient knowledge of the disc/file structure of the other machines. (See Hardware Column page 30.) **A**

## Contact Box

• **Belgian Users** – New User Group in Belgium. For information, contact Philippe Strooband, Meidoorn-laan 17, 1851 Grimbergen.

• **Italian Users** – Contact Guido Masoero, Via Ponchielli 26/C – 10154 Torino.

• **FORTAN Users**, for mutual support and ideas, contact Leslie Pettit, School of Chemistry, The University, Leeds, LS2 9JT. **A**

### COMPUTERWARE

#### SPECIAL OFFERS

CW

- 4 Slot Archimedes Backplane complete with fan etc. £59.95
- 20Mb Hard disc podule £399.95
- Podule less drive £249.95

All Computerware products come supplied with full instructions and mounting metalwork. Prices shown are fully inclusive UK.

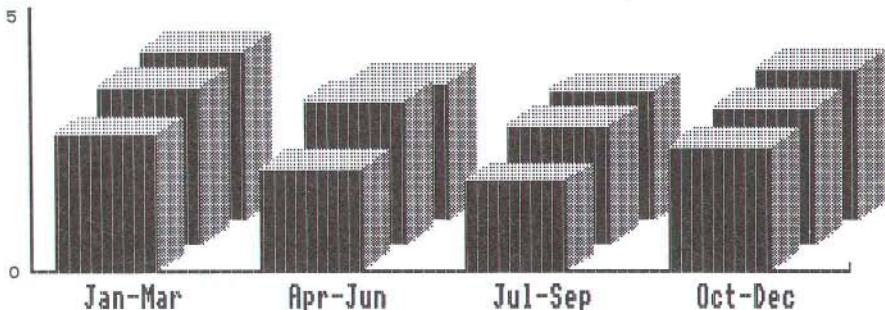
Please allow up to 28 days delivery. Send S.A.E. for details.

order now from: 55 Romulus Court,  
Justin Close, Brentford Dock,  
Brentford, Mddix,  
TW8 8QN.

offer valid for limited period only

# Archimedes

**PRESENTER** is a truly professional graphics program which allows the user to create, modify, print out or photograph high quality colour displays of data in either bar pie or line format. Data is entered either manually in spreadsheet fashion or can be imported from other spreadsheets or programs in comma separated format. **PRESENTER** makes full use of the **Archimedes**' WIMP environment and high resolution graphic modes and incorporates 3D displays, auto-scaling axis, user definable layouts and multi-layer graphs. Screens may be saved for use with graphic packages like **ARTISAN** or graphic wordprocessors like **1ST WORD PLUS & GRAPHIC WRITER**. **PRESENTER** can also be used with **PIPEDREAM**.



At only £24.95 ex-VAT, **PRESENTER** is an invaluable extension to existing spreadsheets and wordprocessors. It is a **must** for all Archimedes' users.

PRESENTER is available from all good Acorn dealers or direct from LINGENUITY.

```

Please send me...PRESENTER Packs      Retail Price    £24.95   £.....
                                           VAT            £ 3.74   £.....
                                           Postage UK     £ 1.15   £.....
                                           Postage Overseas £ 6.00   £.....

                                           TOTAL        £.....

Payment   Visa...   Access...   Cheque...

Credit card number [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]
Card Expiry Date ... / ...
Name(as on credit card) .....
Address .....
.....
.....Postcode.....

Send to LINGUINITY. Dept.JT. P.O.Box 10, Halesworth, Suffolk IP19 ODX.
```

Telephone orders accepted - ring 098 685 476



# Who will venture Forth?

## Karl Strickland

Although Acorn have made available a number of language compilers for the Archimedes, there are still some popular languages that they haven't released. To the rescue comes Silicon Vision Ltd (who have taken over BlueGrey Software's interest) who have produced a Forth compiler for the Archimedes – RiscForth. (They are also working on Modula-2 which will be good news to those who were disappointed when Acorn announced that they wouldn't be releasing their version which requires at least 2 Mbytes of RAM.)

This review concerns itself with RiscForth – first of all, I will say a bit about the language and then go on to discuss the implementation. When you come to learn Forth for the first time – even if you have experience of other programming languages – it will probably not be what you expect. Forth is a different type of language to most others, in that it is a 'stack based' language whereas most other languages such as BASIC, C, Pascal and Modula-2 are all Algol type languages.

Forth was developed at around the same time as BASIC by Charles Moore because he was dissatisfied with the time it took to develop an application in a language such as Fortran or Algol. In contrast to most other languages, Forth can be readily extended and adapted to the user's needs. Forth programs tend to look quite complicated when compared to say a Pascal program because all calculations are done using 'postfix' notation (e.g. `2 3 5 + +` also known as RPN) instead of the more familiar 'infix' (e.g. `2+3+5`). Use of this notation makes the compiler's job easier and also makes the brackets often used in algebraic equations redundant. The Forth language is a bit like 'C' in the sense that it takes time to learn, and the programs sometimes look very cryptic (although much depends on the programmer) but on the other hand it is very powerful.

## Implementation

The implementation is based around the latest Forth-83 standard which is a superset of Forth-79 (anyone remember FigForth on the old Jupiter

Ace?!). All the features detailed in the standard are here together with some special features to take advantage of the Arc. As with Acorn's C, support for graphics and the WIMP system is provided, but so is support for the sprite system. There is also an in-line assembler (like in Acorn's Pascal) which is necessary because you cannot link in ARM code as you can with Acorn's compilers. However, the assembler takes a bit of getting used to. In the Forth tradition, the instruction is entered backwards. First of all, the operands are entered, then the 'S' flag, then the conditions and lastly the operation. So `ADDNES R0,R1,R2` would be written as `R0 R1 R2 S NE ADD`, which isn't too bad to read but a bit of a pain to write!

Also, you get a single-step debugger which can display the state of the stack after each instruction has been executed. Whereas a debugger is counted as a 'luxury' with most languages, this type of debugger is a necessity with Forth – however, this one seems to be quite good and can also be used to good effect when trying to learn Forth because you can see the state of the stack after each word.

A screen editor is included to edit 'screens' (as well as a line editor). If you're a Forth programmer, you will know that screens are what Forth uses to store programs and data. The editor does the job OK, but there is no way to delete the character under the cursor, as you can with <delete> on an IBM PC, or <shift-delete> on the ARM BASIC Editor. All the commands are done using the <ctrl> key – but we have 12 function keys that are unused! Unlike most Forth implementations, it is possible to write and maintain your source code using any text editor and load the resulting ASCII file into RiscForth. TWIN is also supported which is useful because it means that we can use the same editor for all the languages.

Access to the OS command line (OSCLI) is possible and calls to the OS can easily be made via SWIs.

## Multi-tasking is here!

Now for the best feature – Multi-Tasking. This is a first for an Archimedes language. It is important to realise that the multi-tasking is implemented at language translator level as opposed to operating

system level (Arthur 2 and Impulse – where are you?). This means that you can multi-task two or more RiscForth programs together. Although inevitably running more than one task slows each task down, with the speed of the Arc, this only becomes a problem if you are running a large number of processor bound tasks. This is certainly a very welcome addition and this alone might cause people to look at the language who have not previously been interested in Forth. For the technically minded, the multi-tasking uses the round robin technique, but (not surprisingly) dynamic scheduling is not supported (i.e. RiscForth does not change task priorities automatically). However, the programmer can, in effect define the priorities when designing each task.

Floating point arithmetic is supported and of course a Floating Point co-processor will be used if available. Because the ARM is a 32-bit processor, all 'objects' on the stack are treated as 32-bit, be they single/double precision integer or floating point numbers. This does away with the need to have commands to manipulate floating point numbers on the stack – i.e. FOVER, FDUP etc.

Making a stand-alone program with RiscForth is not as straight forward as it is with other compilers. There are a numbers of ways of doing it, but basically, you have to PROTECT the dictionary, re-direct some vectors and \*SAVE a block of memory. But the manual explains it all quite clearly and in practice it is easy to do.

## The Documentation

The manual gives a description of each word in the Forth-83 standard and also, as you would expect, covers all the details of the extensions – the editor, assembler, debugger, multi-tasking etc. The manual does not attempt to teach you Forth but recommends several books that do. The manual is generally well written and informative, but it is sometimes difficult to see where one section stops and the next starts – the headings for each sub-section should be made clearer – perhaps with a bold font.

## Conclusion

If you're already a Forth programmer and want a compiler for the Archimedes – then buy this. It's a good implementation with enough extensions to do useful work on the Archimedes. If you don't know Forth, then buy a book first and have a look at the language. I'll still do my programming in C but that's because of the language and not the implementation. Having said that, C does not have any multi-tasking facilities and if I ever need these, I would have no hesitation in using this. **A**

*Because of limitations of time, we didn't get a chance to show this review to Silicon Vision before publication, but they did make the point that Forth, amongst other things is ideally suited to process control applications, especially with the availability of multi-tasking. Also, it is not a 'toy' language (my word) – it can be, and is, used for real applications. In fact, Silicon Vision have used Forth for writing their schematics capture system for ArcPCB. Ed.*

## Ovid Toolkit Module

- MEDIT – memory editor
  - DEDIT – disc editor 800 and 640k
  - DIS – disassembler
  - DSIZE – shows size of disc
  - SHIFT – shifts memory
  - FILLB – fills memory with a byte
  - FILLW – fills memory with a word
  - COMP – compares two memory areas
  - MFINDSCS – finds a string\* in memory  
(case sensitive)
  - MFINDSCI – finds a string\* in memory  
(case insensitive)
  - MFINDW – finds a word\* in memory
  - DFINDSCS – finds a string\* on disc (case sensitive)
  - DFINDSCI – finds a string\* on disc  
(case insensitive)
  - DFINDW – finds a word\* on disc.
- (\* All these can have wild-cards.)

Send a cheque/P.O. for £15 to:

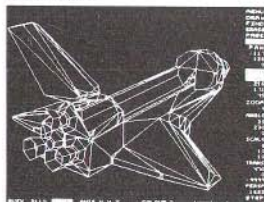
P.A.J.Taylor • 23 Limes Avenue • Alfreton  
Derbys • DE5 7DW



# SILICON VISION

SOFTWARE FOR THE ARCHIMEDES & BBC

## SolidCAD



The ultimate 3D Draughting System for Architectural design, Interior design, Engineering Design and Teaching CDT. Allows drawing in plan, front & side elevations and also directly in 3D view. Includes powerful zoom & pan options for precision draughting and surface definition for creating solid colour objects. Also includes Sweep, Extrude & Macro facilities for designing very complex objects easily. Designs created with SolidCAD are compatible with the Realtime Graphics Language for high-speed flicker-free animation. The custom Archimedes version also performs smooth shading for realism. SolidCAD(Arc) users can upgrade to the Realtime Solids Modeller (Arc) for £40.00.

£49.95 (ARC & BBC) New

## REALTIME SOLIDS MODELLER

The package includes both the sophisticated design environment of SolidCAD and the high speed animation capability of a Realtime Graphics Language (RGL) module developed in pure ARM Risc code for supercharged performance. The package is ideal for Architectural design, Interior design, Engineering design & teaching CDT. The RGL module can be used to create standalone flicker-free animation of designs from your own programs. Smooth shading is also performed for realistic images. Through our in-house expertise in 3D Design and High-speed techniques, no other package can rival the design environment & animation speed of the Realtime Solids Modeller.

£89.95 (ARC) New

## REALTIME GRAPHICS LANGUAGE

The Realtime Graphics Language rom provides a complete 3D Solids Wireframe animation system with 52 star commands and 3D Editors for designing objects to animate from your own programs. Includes a 35,000 pixels/sec line generator for fast 3D drawing rates. 3D Rotate, Scale, Orbit, Perspective and Turtlegraphics. Also compatible with designs created with SolidCAD (BBC).

£49.95 (BBC)

## SUPER-DUMP

The ultimate printer driver which takes advantage of the highest resolution capability of ordinary Epson compatible printers to provide 1920 x 1024 resolution. Images can also be scaled, positioned and previewed before printing. Fully compatible with SolidCAD, Realtime Graphics Language, Gate Array design system & 3D CAD Animation system. Your own graphics programs or other CAD packages can be made compatible with Super-Dump by the addition of a few simple commands. An example program is included in the package.

£15.95 (BBC), £24.95 (ARC) New

## Presentation Manager

The package provides an interactive environment to create, edit and play-back computer controlled presentations for lectures and demonstrations. Also handles graph plotting for polynomials & user-defined functions which can be incorporated within the presentations.

£34.95 (BBC), £49.95 (ARC) New

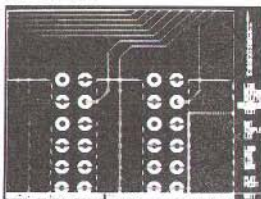
SILICON VISION LTD, SIGNAL HOUSE, LYON ROAD, HARROW  
MIDDLESEX HA1 2AG. TEL: 01-422 2274 or 01-861 2173  
FAX: 01-427 5169. TELEX: 918266 SIGNAL G.

☐ (Access/Mastercard/Eurocard accepted)

All prices include VAT and Carriage (Overseas orders add £4).

SEE US ON STAND 34 AT THE MICRO USER SHOW

## ARC-PCB



The ultimate PCB design system developed specifically for the Archimedes with a specification that cannot be matched. Includes Automatic routing, Rats-nesting, 8 layers, Surface mount capability, 0.001 resolution, 32 x 32 maximum board size, On-line Help, Fast Zoom-Pan-Redraw, Text & Silkscreen facility, Variable Line Pad Text Grid sizes, Part Libraries, Block Move/Copy, Rotate Mirror/Erase options, and up to 300,000 components. For hardcopy, the system supports a large number of plotters and ordinary Epson compatible printers at very high resolutions (1920 x 1024) of 240 dots/inch for near laser quality output. An entry-level ARC-PCB system (without auto-routing) is available for £99.95 (Auto-routing upgrade: £100). Enquire about our turnkey PCB design systems complete with Colour Archimedes and/or plotters with prices from £1220 to £3909 (Inc VAT).

£195.00 (ARC) New

## RiscBASIC

Supercharge your Archimedes Basic programs by compiling them automatically into pure ARM Risc code with the RiscBASIC compiler. Features include Relocatable modules, Cross reference of all variables, functions, and procedures, Floating point and Integer support, Stand alone code generator, Optimising compiler & Full runtime error handler.

£99.95 (ARC) New

## RiscFORTH

A new 32-bit implementation of the FORTH-83 standard, designed to take full advantage of the ARM architecture. Features include Multi-tasking, Optimising compiler, built-in ARM assembler with floating point mnemonics, built-in Fullscreen Editor, File system interface, OS calls support, Floating point & Integer maths, WIMP support, Single-step debugger, Shadow screen for documentation, Block manipulation, Dictionary & Vocabulary display, Call finding and a standalone code generator.

£99.95 (ARC) New

# Matters Arising

• **First Fonts** – We dropped a major clanger here! Apologies to Ian Copestake. Last month, we said that the current version of First Fonts for First Word Plus would only work on Epsoms compatibles and there was nothing available for 9-pin printers. Rubbish! The Maths 5 disc that we sell has versions that work on: Epson LQ printers, Citizen HQP-40, NEC P2200, NEC P6 plus and P7 plus and also Star LC24-10. What's more, Ian is working on implementing it on a number of other printers including the Panasonic and Canon 24pin printers and if yours is not already included, he's prepared to see if it could be. You can't say fairer than that!

What about the 9-pin printers? No, they are not forgotten – Ian has versions working on Star LC10 (called Maths 4) and the Kaga Taxan printers.

He also has greek character sets, Irish Gaelic, German... In fact if you are in need of any different character sets or special characters, ask Ian!

• **Author comes to light** – At last, we've found out who sent the article in the July issue about changing the colour palette on First Word Plus. It was David Scott of Hazel Grove, Stockport. Thanks David!

• **Screen banks on the 440** – There seems to be a problem with the screen banks on the A440. William Doggett's Mandelbrot program (issue 7) doesn't work properly on an A440 – something to do with the screen banks. Has anyone else had problems with programs using screen banks on a 440?

• **Evaluating  $\pi$**  – October issue, page 41, we lost an 'X' in the program listing. The second half of line 90 should read `X% (L%+2) , Y% (L%+2) .`

• **Printer Driver for View B3.0** – Three mistakes in the listing, I'm afraid. The # signs got missed out on two lines: `1410 LDX #poundcode MOD256` and `1420 LDY #poundcode DIV256` and `1540` contained illegal `6502` code – you cannot `LDX (&70),Y!` You have to use: `1540 LDA (&70),Y:TAX`

• **Matrix routines** – one or two folk made comments about Steve Drain's article last month. For example, Andy Pole writes...

The use of the RETURN modifier to an array parameter in a procedure call is redundant and may be misleading. In fact, BASIC V does not create a local copy of array parameters – they are actually passed to a procedure by address. So any alteration made to an array passed by a procedure will also affect that array in the main program.

The claim that 'the inverse of a matrix cannot generally be calculated analytically' is definitely not true. There are well defined algorithms for computing matrix inverses. Perhaps Steve was referring to the numerical stability of such algorithms when implemented on computers. (The comments continue but become far too technical, I suspect, for 98% of us, but Andy has sent in some mathematical routines with a README file which we have put on the Shareware Disc N°2.)

• **MSDOS Column** – Andy also writes about the batch file to reset the prompt and function key 1 saying that it requires the CONFIG.SYS file to contain the following line:

`DEVICE=ANSI.SYS`

in order that the ANSI escape sequences 'e[...]' can be properly interpreted.

• **Disc drives and interfaces** – The moral is, beware when one manufacturer says that the fault lies with the products of a rival company. Obvious, I suppose, but thanks to Vic Budd for reminding me. He goes on to say that he has a Watford 5.25" (Mitsubishi) drive and a Beebug interface and they are working perfectly including on MS-DOS. It may well have something to do with the link settings of the disc drive and also the configuration of the Archimedes.

Actually, Karl Strickland says almost exactly the same. His set-up is fine. Perhaps I ought to put Messers Palmar and Oram in contact with Messers Budd and Strickland!

• **Matrix 3** – Sorry, I got the price wrong on the order form. The price I quoted was EX-VAT, so it should be £100. **A**



# A Textdump Module

## Adrian Look

Here is a little utility that may prove useful. This module prints the current text window (in any mode except 7, and will only send characters to the printer (no escape codes) – so it is compatible with almost every printer! The drawback with this is that it reproduces 20 column modes, 40 column modes, etc all with the same size text. However, if you are using the module for a particular mode then you can set-up your printer to print double-width, normal, or elite depending on which mode you are in.

The module could easily be customised for individual printers and personal preferences. It could even be made to print-out mode 7 screens but that is left up to you (well, we can't do everything)!

Anyway, I shall discuss some of the operating system calls that I used to make up this module.

### Reading the Machine State

First, the module reads the state of the machine: the cursor position, text window, and VDU 2 state, so that it can leave the machine as it found it.

Reading the cursor position is done using OS\_Byte 134, which returns the current x,y co-ordinates of the text cursor, e.g.

```
SYS "OS_Byte",134 TO ,x,y
```

The current text window doesn't have a call all to itself. You have to use OS\_Byte 160, which allows you to read the VDU variables. OS\_Byte 160 can access fifteen of these. You tell the computer which variable you want to read and it will fetch the values of the variable specified and the next variable, i.e.

```
SYS "OS_Byte",160,<variable number> TO  
,<variable value>,<next variable value>
```

Here is a list of the variables: (ic means internal co-ordinates: the origin is always bottom left of the screen and one unit is one pixel wide and high and ec means external co-ordinates: the screen is 1280 units wide by 1024 units high in all modes except mode 22, where it is 976 units high.)

- 0 LSB of graphics window left column (ic)
- 1 MSB of graphics window left column (ic)

- 2 LSB of graphics window bottom row (ic)
- 3 MSB of graphics window bottom row (ic)
- 4 LSB of graphics window right column (ic)
- 5 MSB of graphics window right column (ic)
- 6 LSB of graphics window top row (ic)
- 7 MSB of graphics window top row (ic)
- 8 Text window left column
- 9 Text window bottom row
- 10 Text window right column
- 11 Text window top row
- 12 LSB of graphics origin (ec)
- 13 MSB of graphics origin (ec)
- 14 LSB of graphics origin (ec)
- 15 MSB of graphics origin (ec)

This is only one method of reading the VDU variables – 'OS\_ReadModeVariables' and 'OS\_ReadVduVariables' allow you to obtain much more information about the screen and VDU set-up (see the PRM).

The module also reads the VDU2 state so that it knows whether to leave the printer on or off when it has finished printing out the text dump. This is done by using an OS\_Byte 117, which will tell us about the status of the VDU drivers.

```
SYS "OS_Byte",117 TO ,flags
```

where the bits of the flag variable are:

#### bit meaning

- 0 Printer output enabled by VDU 2
- 2 Paged scrolling selected by VDU 14
- 3 Text window in force (i.e. software scrolling)
- 4 In a shadow mode
- 5 In VDU 5 mode
- 6 Cursor editing in progress
- 7 Screen disabled with VDU 21

### Reading the Contents of the Screen

Having read the initial state of the machine, the module now proceeds to read the contents of the screen. It does this by scanning each text line, with the text cursor, and reading which ASCII character is at the cursor. OS\_Byte 135, allows us to do this.

## Textdump Module

It returns both the ASCII value of the character (0 if there is none) and the screen mode.

i.e. SYS "OS\_Byte",135 TO ,ascii,mode

### Error Trapping

Now all the module has to do is to check whether anything 'goes wrong' while it is printing. The two conditions it checks for are: the escape key being pressed and the printer not responding.

Checking the escape condition is very simple, you just use OS\_ReadEscapeState. If the escape key has been pressed then the SWI will set the carry flag. This is then our signal to stop printing.

For example:

SYS "OS\_ReadEscapeState" TO ,flag%

IF (flag% AND %10)>0 THEN PRINT "Escape"

However, when I wanted to check the whether the printer was on-line, using the routine in Archive1.8 page 7, I got no response! I couldn't get the routine to work as module code. If anyone knows why, may be they could drop me a line?

Anyway, I ended up having to tackle the problem from a different angle. Instead of reading the printer acknowledge line I had to read the printer buffer state to see if the printer was drawing characters from the buffer. If it didn't draw any characters from the buffer within a certain time limit then I generated a 'Printer timed out' error, just like the \*HARDCOPYFX command.

```
10 REM >DumpSrc
20
30 REM *****
40 REM *      TextDump Module      *
50 REM * by Adrian Philip Look *
60 REM * 16th September 1988 *
70 REM *****
80
90 MODE 0
100 PROCassemble
110 OSCLI("Save TextDump "+STR$~
        (code%)+ " "+STR$~(0%))
120 OSCLI("SetType TextDump FFA")
130 END
140
150 DEFPROCassemble
160 DIM code% &4000
170 FOR opt=4 TO 7 STEP 2
180     P%=0:0%=code%
190     [OPT opt
```

```
200
210 .header
220 EQU D 0
230 EQU D 0
240 EQU D 0
250 EQU D 0
260 EQU D title
270 EQU D help
280 EQU D helptable
290 EQU D 0
300 EQU D 0
310 EQU D 0
320 EQU D 0
330
340 .title
350 EQU S "TextDump":EQU B 0:ALIGN
360
370 .help
380 EQU S "TextDump"+CHR$(9)+"1.00
        (16 Sep 1988)":EQU B 0:ALIGN
390
400 .helptable
410 EQU S "TextDump":EQU B 0:ALIGN
420 EQU D start
430 EQU D 0
440 EQU D syntax
450 EQU D help_report
460 EQU D 0
470
480 .help_report
490 EQU S "**TextDump dumps the
        current text window on to
        the printer."
500 EQU W &0D10
510 .syntax
520 EQU S "Syntax: TextDump"
530 EQU B &00
540 ALIGN
550
560 .start
570 STMF D R13!,{R1-R11,R14}
580 ; read which mode we are in
590 MOV R0,#135
600 SWI "OS_Byte"
610 CMP R2,#7
620 BEQ nomodeseven
630
640 ; check whether VDU2 has
        already been performed
650 MOV R0,#117
660 SWI "OS_Byte"
670 AND R1,R1,#1
680 EOR R1,R1,#1
690 ADD R1,R1,#2
700 MOV R9,R1
710
```



720	; read present cursor position	1270	BL restore
730	MOV R0,#134	1280	LDMFD R13!,{R1-R11,PC}
740	SWI "OS_Byte"	1290	
750	MOV R10,R1	1300	.restore
760	MOV R11,R2	1310	STMFD R13!,{R0-R11,R14}
770		1320	; restore cursor position
780	; read maximum x & y positions	1330	MOV R0,#31:SWI "OS_WriteC"
790	MOV R0,#160	1340	MOV R0,R10:SWI "OS_WriteC"
800	MOV R1,#8	1350	MOV R0,R11:SWI "OS_WriteC"
810	SWI "OS_Byte"	1360	; restore VDU2 status
820	MOV R5,R1	1370	MOV R0,R9:SWI "OS_WriteC"
830	MOV R6,R2	1380	LDMFD R13!,{R0-R11,PC}
840	MOV R1,#10	1390	
850	SWI "OS_Byte"	1400	; errors
860	SUB R5,R1,R5	1410	.nomodeseven
870	SUB R6,R6,R2	1420	BL restore
880		1430	ADR R0,error0
890	; set cursor to top of text window	1440	B error
900	MOV R0,#3:SWI "OS_WriteC"	1450	
910	MOV R0,#30:SWI "OS_WriteC"	1460	.offline
920		1470	BL restore
930	; print-out textwindow	1480	MOV R0,#21
940	MOV R3,#0	1490	MOV R1,#3
950	MOV R4,#0	1500	SWI "OS_Byte"
960	.loop	1510	ADR R0,error1
970	MOV R0,#3:SWI "OS_WriteC"	1520	B error
980	MOV R0,#31:SWI "OS_WriteC"	1530	
990	MOV R0,R3:SWI "OS_WriteC"	1540	.escape
1000	MOV R0,R4:SWI "OS_WriteC"	1550	BL restore
1010		1560	MOV R0,#21
1020	;read and print which character	1570	MOV R1,#3
1030	BL checkprinter	1580	SWI "OS_Byte"
1040	MOV R0,#2:SWI "OS_WriteC"	1590	ADR R0,error2
1050	MOV R0,#135	1600	B error
1060	SWI "OS_Byte"	1610	
1070	MOV R0,R1:SWI "OS_WriteC"	1620	.error
1080	MOV R0,#3:SWI "OS_WriteC"	1630	LDMFD R13!,{R1-R11}
1090	BL checkprinter	1640	SWI "OS_GenerateError"
1100		1650	LDMFD R13!,{PC}
1110	; check escape condition	1660	
1120	SWI "OS_ReadEscapeState"	1670	.error0
1130	BCS escape	1680	EQUd &80000001
1140		1690	EQUs "Can't Print Mode 7 screens"
1150	; update cursor position	1700	EQUB 0
1160	ADD R3,R3,#1	1710	ALIGN
1170	CMP R3,R5	1720	
1180	BNE loop	1730	.error1
1190	MOV R3,#0	1740	EQUd &80000001
1200	MOV R0,#2:SWI "OS_WriteC"	1750	EQUs "Printer time out"
1210	MOV R0,#13:SWI "OS_WriteC"	1760	EQUB 0
1220	MOV R0,#3:SWI "OS_WriteC"	1770	ALIGN
1230	ADD R4,R4,#1	1780	
1240	CMP R4,R6	1790	.error2
1250	BNE loop	1800	EQUd 17
1260		1810	EQUs "Escape"

# Hardware Column

## Brian Cowan

This month we are going to consider mainly some aspects of discs, both floppy and hard, and at the end a few words about memory expansion.

### Floppy Discs – Formatting Conventions

There are a number of different sizes of floppy discs currently in use in computer systems. The original discs were eight inches in diameter but these are, thankfully, now obsolete. The three common disc sizes are 5.25, 3.5 and 3.0". The 5.25" standard was, in the past, popular mainly because IBM adopted this for their personal computers. Of course the old BBC micros used this size as well. The 3.0" discs were used mainly on the older Amstrad Z80 based machines and need concern us no further. All the rage is the newest disc size of 3.5" diameter. The Archimedes uses these, the Electron used them, and many of the newer IBM PC clones have gone over to this standard, as well as the new IBM range. Also the Apple Macintosh uses 3.5" discs.

Unfortunately it does not end there. For a given disc size, there are different ways of encoding the information. All methods of storage involve forms of frequency modulation. The oldest method, is known simply as frequency modulation, or FM. It is also referred to as "single density". Subsequently it was realised that a modified form of frequency modulation, MFM, permitted almost a doubling of the amount of data stored. Thus MFM is also called "double density". Another modification permitted a further almost doubling of capacity called MMFM or quadruple density.

Even this is not all! There are different numbers of tracks on discs, different numbers of sectors etc., etc. IBM have their standards, Acorn have theirs, and Mac are a complete law unto themselves.

### What can the Archimedes do?

So why should we be interested in all this? The main reasons relate to questions of compatibility between machines, particularly as the Archimedes is capable of software emulation of other computers. Already we have the PC emulator and of course there is the old 6502 BBC emulator. (Wouldn't it be nice if someone produced a Mac emulator.) Precisely what can be done by the Archimedes depends upon the machine's hardware; in particular, upon the disc controller chip and upon the drive mechanism.

The disc controller chip in the Archimedes is the 1772, similar to that used in the old BBC machines (except the original model A and B). The important point is that this supports FM and MFM encoding only. It will not perform MMFM.

### Old BBC Formats and MSDOS

Clearly, then, the Archimedes can cope with the old BBC disc formats. While firmware is provided for ADFS, the same is not true for DFS. So if you want to read or write DFS discs you must write or purchase software to do this. The IBM formats of 360 kbytes on 5.25" discs and 720 kbytes on 3.5" discs may be handled by the Archimedes, as well as the slightly non standard (?) 720 kbyte 5.25 standard. This is all supported on the MSDOS of the PC emulator. However I don't believe that the

...from previous page...

```
1820 EQUB 0
1830 ALIGN
1840
1850 .checkprinter
1860 STMF D R13!, {R0-R11,R14}
1870 MOV R7,#0
1880 .repeat
1890 ; check escape condition
1900 SWI "OS_ReadEscapeState"
1910 LDMCSFD R13!, {R0-R11,R14}
1920 BCS escape
1930 ; read buffer state
1940 MOV R9,#&16
```

```
1950 MOV R1,#3
1960 MOV R3,#0
1970 CMP PC,#0
1980 SWI "OS_CallAVector"
1990 ADD R7,R7,#1
2000 CMP R7,#&8000
2010 LDMEQFD R13!, {R0-R11,R14}
2020 BEQ offline
2030 CMP R1,#0
2040 BEQ repeat
2050 LDMF D R13!, {R0-R11,PC}
2060 ]
2070 NEXT opt
2080 ENDPROC A
```



emulator software can cope with the 800k DOS format as used by the Master 512 under DOS plus.

The original BBC DFS filling system used FM on 5.25" discs and the user had the choice of 40 or 80 tracks. A single sided 40 track disc could hold all of 100 kbytes! With an 80 track double sided disc this increased to 400 kbyte. ADFS followed, using MFM and usually 80 track double sided discs, the so called L format. The unformatted capacity of such discs was one Mbyte and, when formatted, 640k could be stored. IBM followed a different route. Their original standard was 40 track FM, but with a different sector arrangement from Acorn's DFS.

### Quadruple Density

Unfortunately there is no way that the Archimedes can cope with the newer quadruple density IBM formats. These provide 1.2 Mbytes on a 5.25" disc and 1.44 Mbytes on a 3.5" disc but the encoding is MMFM which is beyond the capability of the 1772.

I think that this is one reason why the proposed but shelved PC podule was to have included a disc controller chip; any modern PC system must cope with quad density discs.

### Mac Disc Format

And so to Macs. What I really would like to be able to do is to write Mac discs on the Archimedes – and maybe even have a Mac emulator. In this way I would be able to prepare text and graphics files on the Archimedes and then load them into a Mac for the secretarial staff to use and print out. Unfortunately this is not to be. Mac discs are completely different. These discs are scanned at constant velocity, so as the head moves in toward the centre the angular velocity increases. The drive mechanisms are therefore different as well. It is simply not possible for Mac discs to be read on an Archimedes or vice versa. (Go on then someone and disprove me!)

### New Winchester Formatter

New 440 machines are coming with a much expanded Winchester formatting program. This is now called HFORM; (Ed has a copy) the old one was called


WFORM. The new program is able to recognise five different drive "shapes" and can reformat the drive accordingly.

This is a somewhat more intelligent program and it will make life slightly easier for those unfortunate people who have to reformat their hard discs. If your disc is not one of four standard shapes – maybe you have installed your own 40 Mbyte drive – then you can choose the disc type OTHER. You are then asked for the specification of the disc drive shape and the disc is formatted accordingly.

### Memory Expansion

Latest projections seem to indicate that the price of 1 Mbit RAMs should be falling by next January. This is just as well because when the new RISC OS 2 is released, multi-tasking will one of its main features. Even the 1 Mbyte of the 310 could be filled when 5 or 6 programs are being run concurrently.

We shall all need to upgrade to 4 Mbytes! I understand that one company is preparing to release a 4 Mbyte expansion board for the 300 machines when the price of RAM chips comes down. We might even see Acorn entering this race! **A**



COMPUTER ASSISTED LEARNING LTD.  
Strathclyde Business Centre,  
Princess Road,  
New Stevenson,  
ML1 4JB.

Dear Barry Christie,

Which 'NEW ART' package for the Archimedes costs under £35 and has the following features:

- \* mode 15 screen with 256 colours.
- \* loads screens from other Archimedes Art Packages.
- \* menu driven for ease of use.
- \* a combination of over 80 different drawing options.
- \* pixel, pattern and text editors.
- \* loads and saves standard or compressed screen files.

Signed

*Art Nouveau*

## ART NOUVEAU

A new Art Package by Barry Christie  
priced £32.50 inc p+p.

Out on 30th November.

# Toolkit Plus from Clares Micros

## Matthew Treagus

Clares Micro Supplies have recently completed their Toolkit Plus that adds considerably to the original Toolkit. Below I have tried to outline all of the major changes between the two versions. (Toolkit was reviewed way back in Archive 1.1, page 22.)

I think many were originally not over impressed with Toolkit but now there is something that really can be used. Memory and disc editors are now in full colour making them more visually pleasing and easier to read. The speed of the internal screen display routines has been improved considerably and all the memory searches have been re-written and their operation is now much more logical. I have no idea why Toolkit bothered displaying the memory it was searching at an unreadable speed, but that has all changed now. The search routines are much faster. One small side-effect of this speed increase is that the escape key is only checked after a match or after the end of a continuous block of memory is reached. This is just a touch annoying to start with but is well worth it for the benefits it gives.

The editors (disc and memory) now contain three extra ctrl keys:

- <ctrl-a> – alter address while in the editor – very useful for A440 owners
- <ctrl-c> – switch case matching on/off within searches
- <ctrl-x> – enables/disables 8 bit mask

The following commands have been added to Toolkit Plus

- Archive & Restore – Backs up a hard disc onto floppies and vice-versa
- Check – Compares all or part of a file with a chunk of memory
- DirAll – Catalogues all directories from a specified path.
- FastLoad & FastSave – Speed improved versions of the OS Screenload/save – They work with full screens only.
- FindAll – Searches all files from a particular point to see if they match certain search parameters such as filename, filetype or file date stamp

Save/LoadPalette – very useful

Tidy – Compacts a disc until there are no memory bubbles, i.e. \*Compacts until there is one continuous area of free space.

IdSwis, SwiName & SwiNo – Are all involved with converting SWI names to numbers and back. IdSwis simply lists all SWI numbers from a specified point.

Other changes: Configuration features more heavily than before. The files archived and searched for are definable in certain file types or date ranges. There are also options to ignore unstamped files. All file types can be entered as number or names e.g. FFB or BASIC.

One of the most powerful commands in Toolkit-Plus is the Archiving command but unfortunately as I am still without a hard-disc and I was unable to test it. The programmer, Mike Shaw, claims the following, of which I have no reason to dis-believe.

Loads/restores to partially corrupted winis/floppies. Multi-copies of the same backup. Extensive error handling and error protection. Selective archiving of certain files by configuration. Large memory buffer from 8k to 3.7MB for faster archiving. Extensive on-screen prompts. Can restore or archive from any winchester position. The archive program is very big and is contained in a separate program on the ToolKit Plus disk. All discs are date/time/volume stamped to avoid problems when restoring. Disc can be formatted will archiving.

## Conclusion

Overall, Toolkit Plus was much more impressive than the original. I would like to see the archiving program changed to allow the backup of networks. However I imagine their would need to be a fair amount of re-programming due to some short comings in the NETFS. Apart from that, it's much easier to use and a more comprehensive set of flexible commands than the original.

Toolkit Plus should be available from the end of October for £49.95 (£46 through Archive.) **A**



# MS-DOS Column

## Ken Biddle

This month we will look at Public Domain Software and Shareware, what they both entail and where you can find out what they have to offer and what you have to pay.

### Public Domain Software

This is basically software that has been written by computer enthusiasts who don't mind letting other people have the results of their labour for nothing. You normally just have to pay a distributor the cost of the floppy disk and postage and you get the software for nothing.

Some of the software is very good quality and some of it not so good, but then it doesn't cost very much (normally just a few pounds) so you haven't lost much if it turns out that you don't like it. On the other hand, as I have said, some of it is very good and you get a real bargain.

### Shareware

This is software that is distributed is freely copyable by anyone for non-commercial purposes. The companies that market the software are licensed to charge a handling fee so that may act as a distribution channel for the software. The principal behind shareware is "Try before you Buy". Most authors supply a full (or Nearly full) version of their program(s) with enough documentation to enable you to adequately use and evaluate the software. If you find that you then use the software and find it good for the purpose that you acquired it then you are asked to support the author (hence "User Supported Software"). You may be asked to send a donation (at the simplest level) and some of the more professional authors can supply full support and manuals for the declared fee. The fees can range from a few pounds to about £80 or £90. In most cases it is around £20 to £30 which in all honesty, for a professionally produced package which some of them are, is quite cheap.

### The Distributors

There are many adverts for both Public Domain software and the Shareware concept and some distributors are better than others. Most will send

you a catalogue of the software they have to offer either for nothing or the price of a stamp. Don't just send off for one catalogue, send off for lots, that way you get the widest possible selection and you also get the best deal. They do not all charge the same for distribution costs, some are much more expensive than others. The more expensive ones do seem to contain larger catalogues than the others but you can often find what you are looking for with the smaller distributors. As a guide, you should expect to pay in the region of £3 to £4 for a 5.25 inch disk and about £4 to £5 for a 3.5 inch disk. (This is another instance where a 5.25 inch disk interface will save you money as well as convenience)

Some of the distributors insist that you join as a member, with others it is optional. The advantages you get by being a member generally is that the price of the individual disks is less.

I have sent off for quite a lot of the most appropriate software from several of the distributors and, in the coming months, I will do some short reviews of them so that you can get a flavour of the package in question, starting with DuraScreen.

### Review of DuraScreen, Release 2.0

DuraScreen is a utility for generating screens in a variety of forms which you may include in your own programs or even Batch files.

It can save you countless hours of program development time. Although the code generated is by no means a total and complete program, it does have the necessary code to test the screen portion. After testing the generated code, you can copy the screen specific code to a disk file for inclusion in almost any application.

Any code generated by DuraScreen is yours to do with as you wish and if you send off for the user supported version (\$10) then you will get the source code (in Turbo Pascal) which you can modify to suit your own needs, but the code to DuraScreen (modified or otherwise) may not be distributed.

DuraScreen generates fancy colour menus and input screens. The number of colours and complexity of the screens generated is quite

extensive. It must be mentioned, however, that a large number and variety of colours will cause your generated program to be larger. A simple one colour program might require 50 lines. A program that generates a rainbow of colours and dozens of screen inputs can generate hundreds of lines of code.

Screens may be generated in the following formats: BASIC screens, C language screens, Assembler screens, Prolog screens, dBase III + screens, Pascal screens, ANSI compatible screens and ASCII format screens.

DuraScreen can also read ANSI files and you can convert them to a Pascal, BASIC A, etc. program. You are also supplied with a utility to load as a TSR (Terminate and Stay Resident) module which is like a relocatable module on the Archimedes. This will take snap shots of screens and save them as ANSI files which you can subsequently load into DuraScreen, modify and/or generate the appropriate code for your own use.

This utility is extremely useful for creating fancy Batch files which look good and are descriptive. The package is excellent value for money for this type of utility.

### Distributors of PD & Shareware

Here is a list of addresses of some of the distributors that you may obtain Public domain and/or Shareware software from :

Shareware Marketing, 87 High Street, Tonbridge, Kent, TN9 1RX.

PC Serve, 1500A Greenford Road, Greenford, Middx, UB6 0HP.

Vasstec Ltd, 4 Royal Terrace, Glasgow, G3 7NT.

Premium Software, 24 Station Road, Barton Halsall, Nr Ormskirk, Lancs. L39 7JN.

Seltec Computer Products Ltd. Northumberland House, Staines Business Centre, Gresham Road, Staines, Middlesex, TW18 2AP.

### 5.25 Inch Disc Drives

Lastly this month I have had some more queries regarding the setting up and use of external 5.25 inch disk drives. Rather than put all of the information in the column again I am producing an MSDOS Facts, Hints and Tips booklet which hopefully should contain all of the above information plus lots of other advice and help. It should be ready in about a month's time and anyone wanting it should just send a large S.A.E. The MSDOS Software compatibility list is now ready (although growing each month) and anyone wanting that should do the same as for the Facts Sheet.

I have had some queries about reading 5.25 inch disks that have been created on an Amstrad P.C. I have since tried moving some software from the Amstrad and have not had any problems. (Bully for me you all say!) In fact you might like to know that Locomotive BASIC 2 works (running under GEM) although painfully slowly.

Next month hopefully some more Hints and tips and in the meantime perhaps I'll see you at the Micro User Show. Bye for now ... Ken. **A**

### Assembly Language – Answer to exercise

The mistake is in lines 180 and 200. When the word being shifted is being written back by the instruction it only needs to be moved eight places each time, not sixteen and then twenty-four! So the corrections are: 180 MOV R0,R0,ROR #8 and the same in line 200. **A**

## Computing and Electronics Holiday

'A residential holiday with a difference' for 13 to 16 year olds  
in the peaceful Hertfordshire countryside. **Nine days for just £76.**

The holiday is run by a team of committed Christians  
under the auspices of 'Scripture Union in Schools'.

Dates: **August 1st – 9th.** For more details, contact...

**Jim Maddox, 69 Claverdale Road, London, SW2 2DH. (01-671-8761)**



# Assembly Language Programming – 5

## Alan Glover

So far, we have kept to the simpler types of MOV instruction such as MOV R0,#34 or MOV R0,R1.

In fact ARM instructions can use three (sometimes four) registers, and usually take the form :

<instr> <destin'n> <l-h data> <r-h data>

After the instruction, the <destination> will contain the result of an operation performed upon <left hand data> and <right hand data>.

MOV is a special case, of the form :

<instr> <destin'n> <right hand data>

In fact there are five different types of <right hand data>, which we look at next:

### Register: e.g. MOV R0,Rn

Rn – A register, which contains the operand

### Immediate: e.g. MOV R0,#n

#n – An eight bit number, which may be mapped to any consecutive eight bits of the 32 bit number by a four bit shift which is automatically encoded in the instruction. However, because only four bits are used (instead of the five needed to represent 0-31), the data bits can only be shifted to even bit positions, e.g. 0th, 2nd 4th, 6th bits, and so on. Thankfully, the encoding of numbers in this manner is handled by the assembler – which will complain if a number uses more than eight bits, or cannot be positioned on an even bit boundary.

### Shifted RRR: e.g. MOV R0,R1,RRX

Rn,RRX – A register, whose contents have been rotated one place to the right, with the old bit 0 going to the carry flag and the old carry flag going to bit 31. Note that this only applies to the value used in the instruction. The register itself is not altered by this.



### Register shifted: e.g. MOV R0,R1,LSL R2

Immediate shifted: e.g. MOV R0,R1,LSL #4  
Rn,LSL #n / Rn,ASL #n / Rn,LSL Rn / Rn,ASL Rn – A register whose contents have been shifted

left by the number of moves given by an immediate value (0-31) or a register (which would contain 0-31). Each bit movement causes 0 to replace old bit 0, and old bit 31 to move into the carry flag. (LSL and ASL are exactly equivalent.)

$C \leftarrow b31 \dots b0 \leftarrow 0$

Rn,ASR #n / Rn,ASR Rn – A register whose contents have been arithmetically shifted right by the number of places given by the immediate value or the register. The range is 1-32. The idea of this shift is that the sign bit is not altered and it works by shifting the word one bit down, with old bit 0 going to the carry flag, but old bit 31 being put back as bit 31. So after the first shift, bit 30 would contain the value from bit 31 and bit 31 would still contain its original value. On the next shift bit 29 would be given the same bit and so on.

$b31 \rightarrow b31 \dots b0 \rightarrow C$

Rn,ROR #n / Rn,ROR Rn – A register whose contents have been rotated by the number of places given in the immediate value or the register (1-31). The carry flag also contains the bit most recently rotated from bit 0 to bit 31.



### What shifts are possible?

At first sight, a shift such as RLX might appear to be missing (rotate left including the carry). However this is in fact just the same as adding the number to itself using the carry flag before and afterwards, i.e. ADCS R0,R0,R0. ADC is Add with Carry, which will be detailed next month.

With a little thought, some quite intricate operations can be performed in one instruction using the shifts.

For example ADD R0,R0,R0,LSL #3 will add R0 to  $R0 \times 2^3$  (8 times). The result is to multiply R0 by nine in one move.

### An example

The following example shows ROR in action. R1 is set to hold the ASCII characters D, C, B and A. These are then displayed. In the first case, the result

of the shift is going into R0, and R1 is unaltered. Consequently the first shift is eight bits, the second sixteen, and so on. The second time through, the rotated value is being written back, so it only needs to be moved by eight each time. Note that OS\_WriteC ignores all except the lowest eight bits.

```

50
60 MOV R1,#&41000000
70 ADD R1,R1,#&00420000
80 ADD R1,R1,#&00004300
90 ADD R1,R1,#&00000044
100
110 MOV R0,R1
120 SWI "OS_WriteC"
130 MOV R0,R1,ROR #8
140 SWI "OS_WriteC"
150 MOV R0,R1,ROR #16
160 SWI "OS_WriteC"
170 MOV R0,R1,ROR #24
180 SWI "OS_WriteC"
190 SWI "OS_NewLine"
200
210 MOV R0,R1
211 SWI "OS_WriteC"
212 MOV R0,R0,ROR #8
213 SWI "OS_WriteC"
214 MOV R0,R0,ROR #8
215 SWI "OS_WriteC"
216 MOV R0,R0,ROR #8
217 SWI "OS_WriteC"
220
230 SWI "OS_NewLine"

```

The method of loading R1 in lines 60-90 is quite long winded and is caused by the method of storage

of immediate values (8 bits, with a four bit shift). There are quicker ways but we haven't covered them yet!

### Next month

The instructions available fall into five groups. In the next couple of parts we begin working through each group, covering all the instructions.

### Your turn

This month's program is intended to make you think a little. It is meant to show "abcd" on the screen, but is currently showing "abdc".

Examine the program, and the information in this part and decide what is wrong and correct it. (Solution on page 34.)

```

50
60 MOV R0,#&64 ;"d"
70 MOV R1,#&63 ;"c"
80 ADD R0,R1,R0,LSL #8
90 MOV R1,#&62 ;"b"
100 ADD R0,R1,R0,LSL #8
110 MOV R1,#&61 ;"a"
120 ADD R0,R1,R0,LSL #8
130
140
150 SWI "OS_WriteC" ;"a"
160 MOV R0,R0,ROR #8
170 SWI "OS_WriteC" ;"b"
180 MOV R0,R0,ROR #16
190 SWI "OS_WriteC" ;"c"
200 MOV R0,R0,ROR #24
210 SWI "OS_WriteC" ;"d" A

```

## WIMP Templates

### Adrian Look

*This month Adrian teases us by showing how we can simplify using the WIMPs by loading the definitions from template files. Then he tells us that we cannot, yet, actual create the templates! Mind you, he holds out hope for us by promising to finish the Template Editor program for us by the end of the month.*

The Arthur Window Manager (AWM) has a facility to load window definitions from a 'template file'. This 'template file' can contain information about

any number of windows (and icons therein) along with the resources that these windows might use.

Archimedes windows use two types of resource: memory (for indirected data – see Archive 1.12, page 30) and anti-aliased fonts. The problem with this is that any references to memory and anti-aliased fonts (via font handles) cannot be guaranteed to be the same when the window definition is loaded by the same or another application program. If this happens then the



Archimedes may crash, or at best not do what you want it to do!

How is this solved? Well, when loading templates, the application program must provide three things – workspace into which the window definition can be loaded, workspace for indirected data and a 256-byte font reference array. This will allow the AWM to recalculate any references to indirected data and anti-aliased fonts.

The first two of these requirements are more or less self-explanatory but I should point out that, the workspace into which the window definition is loaded can be reused. On exit from SYS "Wimp\_LoadTemplate", the window definition is in this workspace. The application program can then decide what it wishes to do the information.

The indirected data workspace, however, must be preserved. This does not mean that the contents cannot be changed, only that the workspace must be used to store the window's indirected data! But, if the window is no longer required then neither is its indirected data workspace.

The third requirement (a 256-byte font reference array) is needed to 'keep tabs' on any anti-aliased fonts used in a window definition. By doing this you can use the font cache as efficiently as possible (by releasing any fonts used in a window, when the window is not being used). If you don't want to use anti-aliased fonts, skip the next three paragraphs as they will probably confuse you more than inform you. In any case, if you don't use anti-aliased fonts in your window definitions, you don't have to worry about the 256-byte reference array.

Each byte of the reference array represents the usage of a particular font in the window definition being loaded. e.g. byte 0 contains the number of times the font with a handle of 0 is used in the window definition. So when the window definition is loaded in, for every occurrence of an anti-aliased font, the AWM calls a SYS "Font\_FindFont" and the relevant byte (denoted by the font's handle) in the array is incremented.

Hence, all 256 bytes of the array should be set to zero before the template is loaded, so that the exit values of the array indicate how many times each font was used. The application program should then

take note of this, so that when the window is closed it can call SYS "Font\_LoseFont" the appropriate number of times.

Another method would be to keep a count of the fonts used by all the windows and only release them when the application is finished. This would be done by resetting the array to zero at the beginning of the application and then using the same array for every template loaded (and not to reset it).

If you are not confused by now, you are doing well! However, although templates are quite difficult to understand they can still be of help to us mere mortals! The thing is, we don't actually have to know too much about how template work, just that they do. The next section illustrates the three SWI routines used for templates by giving two short PROCedures and a FN. Using these we can load a window definition from a file and forget about how it does it.

### 'Wimp\_OpenTemplate' &400D9

This routine will cause the AWM to open a template file and read the header information – i.e. find out what fonts are used and which window definitions are in the file, and whereabouts these definitions are in the file.

Note: the AWM can only access one template file at a time.

```
DEFPROCopentemplate(filename$)
SYS "Wimp_OpenTemplate",,filename$
ENDPROC
```

### 'Wimp\_LoadTemplate' &400D8

Each window definition is denoted in the template file by an identifier. This is a 12-byte string. Once you have set up the definition workspace, indirected data workspace and font reference array (when necessary), you should use this routine to load a particular window definition. Note: the identifier string can contain wild cards. e.g. 'text\*'

```
DIM block% &100
DIM indirect% &100
DIM fonts% 255 : REM if font to be
ignored then let fonts%=-1
:
REM load templates here
```

```
END
:
DEFFNload_template(ident$)
LOCAL next%
SYS "Wimp_LoadTemplate",,block%,
    indirect%,end%,font$,ident$,0
    TO ,,indirect%,,,ident$,next%
IF next%=0 THEN ident$="" : REM if
    not found then make result null
=ident$
```

### 'Wimp\_CloseTemplate' &400DA

This will cause the AWM to close the current template file.

```
DEFFPROCclosetemplate
SYS "Wimp_CloseTemplate"
ENDPROC
```

For example this is how we would load a window definition whose identifier is "tree" from a template file called "windows".

```
PROCopentemplate("windows")
void$=FNloadtemplate("tree")
PROCclosetemplate
IF void$<>"tree" THEN ERROR
1,"Template not found!"
```

The window definition "tree" would now be stored in the workspace called block%, so to create the window all that has to be done is:

```
SYS "Wimp_CreateWindow",,block% TO
                                handle%
```

For those who want to see a full implementation of this, I have put an example program on this month's disk. If you object to buying the program disk, then if you send a blank disk (and an SAE) to Paul, he will gladly put the program on it and return the disk.

As you can see, actually using template files is not that difficult after all. However, there is a snag! Acorn say that there is a template editor available to allow us to create the template files in the first place. This has not been (as yet) been made generally available but I am in the process of turning the semi-working version that Acorn provided into a usable program. So what will it do?

### The WIMP Template Editor

The WIMP Template Editor will contain facilities which will allow you to design every feature of the WIMP environment. This will include various editors for: the palette, sprites, windows (and the icons therein), menus and pointers.

The Template editor is, of course, fully WIMP driven itself and employs a WYSIWYG format – this enables you to completely forget about setting up data blocks for your window and menu definitions and just concentrate on their format.

The Template editor will also come with two extension modules. The first will be the sprite extension module mentioned last month – this is used to provide the Template Editor with a whole host of sprite editing facilities. The second and probably more important will be a module which give you the facility to load menu definitions as well as window definitions from a file – which the present AWM lacks.

When will it be available? Well, I have set myself a dead-line of the 30th November. So by the time you receive the next issue of Archive, the Template Editor should be ready for the public!

Cost? I was hoping to release it as a Public Domain program but this would mean that you (the user) would only get an undocumented semi-useful version. Instead, I decided that I would devote more time and effort into it and sell it at £9.50. (£8.00 to Archive subscribers.) **A**

SEE US AT THE



November 11-13 1988  
New Horticultural Hall  
London

We'll be on stand number 2  
– come and see us!



# BASIC Forum

## Clifford Hoggarth

This month's listing is a checksum utility for BASIC programs written by Clive Payne. The idea of a checksum is to provide a means of ensuring that two sets of numbers are identical. In the case of computers the numbers are usually byte values which may represent text, data or code for software. The most common usage is for confirming correct transmission of data between computers e.g. the CRC bytes of the BBC Micro tape filing system. The use here is to help correct typing of published listings, by comparing the checksum produced from the program with that published. Obviously this only works if the same algorithm is used in both cases - this program merely uses one of many possibilities.

The program prompts for a file name and then, if the file is a BASIC program, calculates a checksum for each line. The checksum itself is merely the low 16 bits of the sum of the product of the ASCII code of each character in the line and its position in the line, e.g. the line: 10 a=1

produces a checksum of:

$$\begin{aligned} \text{ASC("a")} * 1 + \text{ASC("=")} * 2 + \text{ASC("1")} * 3 \\ = 366 \text{ (&16E)} \end{aligned}$$

Note that the program skips blank lines. This is because there is no line to produce a checksum with. One thing the program does not allow for is leading spaces. Hence it is probably best to use it without indentation to avoid problems with extra spaces. This also reveals a difference between the BASIC EDITOR and the command prompt of BASIC V. Blank lines typed into the editor contain no characters but blank lines typed in at the prompt will contain a space (necessary to ensure the acceptance of the line). This is not true, however, if a LISTO option other than 0 is used then leading spaces are stripped from lines typed at the prompt and a blank line will therefore have no length.

A couple of the functions used in this program are useful elsewhere, namely FNfiletype and FNyn.

FNfiletype takes as its argument a filename and returns the file type. It uses a SYS "OS\_File" call

which returns the "load address" of the file which contains the file type in bits 16-24. (N.B. an error will occur if the file does not exist.) These are separated by ANDING the load address with &FFF0 to mask out the date stamp. The value obtained is then shifted to the lowest 12 bits using the >>> function.

FNyn is used to check for a yes or no response to a prompt, which is supplied as the functions argument. Key presses are tested until the <Y> or <N> key is pressed. The value returned by GET is ANDed with &5F so that the routine is not sensitive to case. The value returned is determined by the expression: (temp=&59). This will give the value -1 (TRUE) if temp=&59 and 0 (FALSE) if temp<>&59. The use of comparisons to produce a numerical value in this manner can be incorporated into more complex functions. For example, incrementing or decrementing a value according to a key press can be done with a single calculation:

```
offset =  
offset - (key=ASC("I"))*increment  
+ (key=ASC("D"))*decrement
```

'offset' will be incremented if <I> is pressed (-1\*increment)+0\*decrement) and decremented if <D> is pressed (-0\*decrement+(-1\*decrement)). If neither key is pressed then no action will occur (-0\*decrement+0\*increment). Note the necessity to use the subtraction to increase the value and addition to decrease the value in order to overcome the negative value of TRUE (i.e. -1)

```
10 REM > &.Basic.Checksum  
20 REM To produce a checksum  
30 REM for BASIC programs  
40 REM Written by Clive Payne  
50 REM Adapted by Clifford Hoggarth  
60  
70 MODE 12  
80 ON ERROR PROCerror:END  
90  
100 REPEAT  
110   CLS  
120   hardflag=FALSE  
130   REPEAT  
140     REPEAT
```

## Checksum Program

```

150     INPUT "Program name? "name$
160     IF name$ <> "" THEN
170         channel% = OPENIN(name$)
180         CLOSE#channel%
190     ELSE
200         channel% = 0
210     ENDIF
220     IF channel% = 0 THEN PRINT
230         "File does not exist!"
240     UNTIL channel% <> 0
250     type% = FNfiletype(name$)
260     IF type% <> &FFB THEN PRINT
270         "File is not a BASIC
280         program!"
290     UNTIL type% = &FFB
300     IF FNyn("Hardcopy") THEN
310         hardflag=TRUE
320     channel% = OPENIN(name$)
330     PRINT
340     REPEAT
350         byte% = BGET#channel%
360         line_high% = BGET#channel%
370         IF line_high% <> &FF THEN
380             line_low% = BGET#channel%
390             line_length%=BGET#channel%
400             PROCcheck
410             PROCdisp
420             PROCgetkey
430         ENDIF
440     UNTIL line_high% = &FF
450     CLOSE#channel%
460     UNTIL NOT FNyn("Another file")
470     END
480     DEF PROCcheck
490     LOCAL loop%, char%
500     check% = 0
510     IF line_length% > 4 THEN
520         FOR loop% = 1 TO line_length%-4
530             char% = BGET#channel%
540             check% +=char%*loop%
550         NEXT loop%
560     ENDIF
570     ENDPROC
580     DEF PROCdisp
590     LOCAL line$
600     line_num% = line_low%+(256*line_
610         high%)
620     PROCjustify(line_num%,7)
630     line$ = an$
640     PROCformat(check%)
650     line$ += " "
660     line$ += an$
670     IF hardflag THEN VDU 2
680     PRINT line$
690     VDU 3
700     ENDPROC
710     DEF PROCjustify(amt%,fs%)
720     an$ = STR$(amt%)
730     an$ = STRING$(fs%-LEN(an$)," ")
740     + an$
750     ENDPROC
760     DEF PROCformat(amt%)
770     amt% = amt% MOD &FFFF
780     an$ = STR$(amt%)
790     IF LEN(an$) < 4 THEN
800         an$ = STRING$(4-LEN(an$),"0")
810         + an$
820     ENDIF
830     ENDPROC
840     DEF PROCgetkey
850     LOCAL key%
860     key%=GET
870     ENDPROC
880
890     DEF PROCerror
900     VDU 3
910     CLOSE#0
920     REPORT
930     PRINT" at line ";ERL
940     ENDPROC
950
960     DEF FNyn(temp$)
970     LOCAL temp
980     PRINT temp$"...? ";
990     REPEAT
1000     temp=GET AND &5F
1010     UNTIL temp=&4E OR temp=&59
1020     VDU temp
1030     =(temp=&59)
1040
1050     DEF FNfiletype(filename$)
1060     LOCAL type%
1070     SYS "OS_File",5,filename$ TO
1080     ,type%
1090     type%=(type% AND &FFF00)>>>8
1100     =type%A

```



# ISO-Pascal Extensions (part 2)

**David Wild**

*Continuing from last month's article, David now turns his attention to modules and static variables*

## Modules and Static Variables

This is the most important part of the extensions. The three pages of the manual devoted to these topics introduce a number of features which can make life very different for the pascal programmer.

## Modules

Modules are sub-programs, compiled separately, containing variables and/or procedures which can be used by other modules and procedures. This bald statement hides the fact that you can now write pascal code for general tasks and distribute it without having to give away the secrets of your code. (In these notes, 'routine' is used to mean a function or a procedure.)

You combine modules at link time, so if one of the modules used in a final program needs changing, all that needs to be done is to re-link the program without any need to re-compile all the program code. This is especially useful when one module is used in a number of final programs.

A module looks almost the same as a program, except that at least one of the variables or routines is preceded by the word 'export' and there is no code that is not contained within a routine. There is a restriction that a module cannot use the 'read' and 'write' procedures, except for files, but this will not normally be a problem.

The general form of a module is :-

```
MODULE name;
CONST
TYPE
STATIC, IMPORT or EXPORT variables
LOCAL, IMPORT or EXPORT routines.
END.
```

The normal rules of scope apply, so that no routine can refer to any variable, or routine, outside its normal range except for those explicitly imported.

IMPORT routines, which must be declared as EXPORT in some other module, have the normal

header declaration preceded by the word 'import', but no body – as that is supplied by the module from which the routine is being imported.

EXPORT routines are exactly the same as a normal routine, except for the word 'export' preceding their header.

A module can contain as many routines as may be necessary for the work it is to do, but only those preceded by the word 'export' can be used in other programs or modules. This means that you can arrange for use of the module to be regulated by the export routines you provide and the user does not know about any of the support routines that are used. This is especially useful when the incorrect use of an internal routine could give misleading results. By making sure that these routines can only be called in the right way you may be able to eliminate the use of some checking routines – such as 'divide by zero'.

Now let us look at the way in which modules might be used. We will assume that a statistical module has been provided, that has an export function called 'mean'.

```
module statistics;
export function mean(inarray :
array[1..100] of integer):real;
var
loopcount : integer;
sum : integer;
begin
sum := 0;
for loopcount := 1 to 100 do
sum := sum + inarray[loopcount];
mean := sum / loopcount
end; {I realise that it's not a
very practical function!}
```

In your program you would declare

```
import function mean(inarray :
array[1..100] of integer):real;
```

with no function body, then use it in the normal way.

## Aliases

There could be a problem with importing routines from modules, in that one of the routines to be

imported had the same name as one that you intended to write in your main program, or one to be imported from another module.

Because of this, pascal provides aliases. These can be provided explicitly by the programmer of the module or can be derived implicitly from the routine name. If the programmer provides an explicit alias you **must** use this, if not you can just use the routine name if it suits your purpose.

If the programmer of 'mean' had provided an alias it might well have been like this :-

```
export function mean alias
'average'(inarray : array[1..100]
of integer):real;
```

(note the quotes round average) and you might call it by :-

```
import function mean100 alias
'average'(inarray : array[1..100]
of integer):real;
```

and you would refer to it in your program as 'mean100'. Note that the case of any letters between the quotes is significant.

If no explicit alias is provided then you can use the name of the exported function IN CAPITALS as the alias, like this :-

```
import function mean100 alias
'MEAN'(inarray : array[1..100] of
integer):real;
```

and once again your program would refer to it as 'mean100'.

### Static variables

There are three new types of variable usually, but not necessarily, associated with modules. The three types are STATIC, EXPORT and IMPORT. This last one, as implied, is a variable imported from a module – complete with any value it might have. Apart from the preceding word, 'import', it is declared in exactly the same way as any other variable.

Static and export variables are more interesting. Apart from the fact that a variable to be exported (only inside a module) must have its name preceded by the word 'export' instead of 'static', these two are identical and they have two very interesting characteristics.

Firstly, they hold their values until they are changed by explicit action. This means that a routine can know whether or not it has been called before and so avoid doing initial calculations on subsequent calls. Similarly, counts could be made of the number of times a particular routine had been called. This is similar to the use of a global variable but with the advantage that it cannot be corrupted accidentally in another part of the program.

Secondly, they can be given initial values, which are calculated by the compiler rather than at run time. The manual gives several examples of this, including the initialisation of arrays and then says that you cannot initialise sets, pointers and arrays! My experience suggests that this should read 'arrays of arrays'. In practice you can initialise arrays of integers or reals and you can initialise strings – which are a type of array – but not arrays of strings.

### Documentation of modules

Because other users of any modules that you write have no access to the source code unless you give it to them, you will need to make sure that every exported routine and variable is properly documented with details of any aliases, including the case of the letters in those given explicitly. You will also need to make sure that you have explained the circumstances in which your routines will work. It is reasonable to expect the calling routine to make some checks on the data, just as you expect an error message (and failure) if you try to take the square root of a negative number, but your user must know what the restrictions are. **A**

---

*Continued from page 52...*

vaguely recognisable environment, in fact without having the benefit of a cmos ram saver, the only effective way back to a usable configuration is to use the power-up reset facility which installs a default configuration that is rarely suited to the users requirements.

This piece of software represents a colossal effort – the result of many individuals' work and can be summarised as being highly valuable due to the flexibility of the command structure that allows it to be tailored to any image processing application.

AIM is distributed by Lingenuity at "about £6". **A**



# Arthur – 'C' Clerk

## Tony Still

The C language seems to have been neglected in the pages of Archive in recent months. This item seeks to rectify this to a small extent and is about my personal experiences with the Acorn C compiler and the Arthur CLI (command line interpreter). I bought the compiler with the aim of learning C, never having used it, but building on experience with other high-level languages. Given this starting point, the aim of the following is to talk about the practicalities of C development on the Archimedes. Much of the discussion on Arthur and C is not actually specific to C at all, but is about the efficient use of Arthur for repetitive jobs.

My machine is an Archimedes 310 with colour monitor plus a single floppy drive. I have Arthur 1.20 and C version 1.54A. C requires at least 1 Mbyte of memory.

## The Compiler

The compiler package contains a single disc and a very slim (70 pages) guide to using the system. This means that some other guide to the C language is also needed. I already had a copy of Kernighan and Ritchie ('The C Programming Language', Prentice Hall – the standard guide to C – often called just 'K&R') but a careful read of the Acorn manual's introduction convinced me that this was not going to be adequate.

The Acorn compiler is a full implementation of the new ANSI (draft) standard for C while the original K&R covers the first (de-facto standard) definition of C. The Second Edition of K&R has recently been published which covers ANSI C. This book is the definitive reference for C, and I would consider it essential for anyone who is going to program seriously in C (and why else would you spend £100+ on a C compiler?). Do get the Second Edition though and be aware of the price – my copy cost £24.90 from Foyles.

For an experienced programmer, e.g. in Pascal, K&R is all that is needed but it will not teach C otherwise. C is a professional language with all the power needed to write large applications (like UNIX!) so some of its features are off-putting to the

beginner. K&R make a good attempt at a Tutorial but an easier book is necessary if you are just starting. I have been using "C The Complete Reference" by Herbert Schildt (McGraw Hill 1987) which also contains a much more detailed coverage of the ANSI library (see below); this cost me a further £21.95.

There is one more essential item. The compiler package does not include an editor. I guess that it is possible to use the BASIC editor, and Twin would obviously be ideal but First Word Plus could also do the job. I use a very strange choice because I already had it to hand and that is the BBC ISO Pascal editor running under BBC emulation. If you only have one disc drive then FWP may not be such a good idea because of size limitations (swapping discs for every bug correction gets very irritating).

All this is not to put anyone off using C; the language is a joy to use and Acorn's compiler is a solid product. It compiles quickly (once it has finished loading itself) and is reasonably bug-free. The C package includes extensive libraries for all of the ANSI Standard plus a large Arthur library. In terms of actually controlling an Archimedes it is very similar to BASIC because many of the same facilities are provided by the library.

## How well does it work?

The compiler produces native object code (i.e. real ARM assembler, not some intermediate code which is then interpreted in a similar way to BASIC). From a brief inspection, the object code appears to be of reasonable quality and the performance is good, with optimisation being evident (i.e. the compiler intelligently 'writes' the assembler including use of the ARM's ability to sometimes do two things in one instruction).

The speed of the final code is high for integer-based programs but a little disappointing on floating point. In fact, comparing small C programs to small BASIC programs with extensive floating point gave me little or no improvement, but there are several reasons for this: (1) BASIC is working to less precision than C, so C makes the ARM work harder even if the extra accuracy is not needed and

(2) C is working with a general purpose Floating Point Emulator while BASIC is optimised to its own needs. (This does mean that the C code will eventually be able to use the Hardware Floating Point Co-Processor, then it should really fly!) (3) A small program flatters an interpreter. As the program size grows an interpreter like BASIC will spend more and more time on house-keeping functions, like looking up variable names, thus it will tend to slow down. A compiler like C does all of this house-keeping at compile-time (i.e. before the program runs) so its speed is independent of program size. Similarly, large comments or long names will have no effect on C's speed.

It is also worth remembering just how good the BBC BASIC interpreter really is, which gives even compilers something to live up to.

To quantify the real speed of any language is difficult and a number of benchmarks have already been published for C. (e.g. A & B Computing, January 1988) The simple result which impressed me was the time taken for one million iterations of an empty 'for' loop... fractionally under one second! (Compare this to over 14 seconds for BASIC, but the test does flatter the compiler in this case).

The language actually accepted by the compiler is the 'new' ANSI standard. This has several new features compared to original C while retaining upward compatibility. The differences are listed in K&R (2nd edition!) but there is one major improvement:

It is now possible to declare a function (procedure) 'prototype', at the head of the program, for each function. This enables the compiler to check the consistent use of both the number and type of parameters, enabling many trivial errors to be found at compile time.

### Using the Compiler

It is quite possible to use the compiler with a single disc drive but it does require some organisation. The obvious first step is to write protect the C master disc and make a working copy of it (via \*Backup). I made two copies of my disc and then hid it well away - I've never spent that much of my own money on one disc before!

To really work efficiently, everything needs to be on one disc, so the next thing to do is copy your

editor into \$.Library on the copy C disc (in my case I also needed the BBC Emulator). The compiler, linker and floating point emulator are already there.

The compiler manual hints at the built-in filing system in the compiler which in fact contains copies of the standard header files; these files describe the run-time library to the compiler. The important point is that they are large text files which don't actually need to be on the work disc. Thus your next stop should be \$.ARM.CLIB.H where you can delete Arthur and stdio - the rest are small by comparison but these two will save over 30k.

Once the disc contains all of the standard files, the current directory should be set using \*DIR \$.ARM.BENCH (or File\$Path could be updated to include this). The compiler assumes this and looks for source files in subdirectory C (so the editor will see the file as 'C.File' while the compiler just calls it 'File'). This \*Dir is always needed so it is advisable to put it into the !Boot file.

### Working Efficiently

It is often said that a lazy programmer is a good programmer since he or she will always look for the easy way around a problem. I qualify on the first count, but also my typing is slow so I rapidly tired of all the commands needed to get from bug, to editor, to recompile and rerun. This is where Arthur can become the C Clerk of the title.

The Command Line Interpreter allows symbols (names) to be defined as some arbitrary string which will then be substituted when used as a command parameter. Select \*Dir \$ and then try:

```
*SET Fred Library
*. <Fred>
```

This should result in a catalogue listing of the library directory. In general \*SET allows any name to stand for any other, the <> causing the translation to happen. This is the key to saving typing. So, at the start of a C session, we need something like:

```
*SET C$File TheFileToWorkWithToday
```

A set of utility macros can then be written which use C\$File to choose the files to work on. (There is nothing magic about the name or the \$ sign).

There is a problem with this - the compiler will not accept translations of this form on its command



line; in fact it mistakes the < for an attempt to redirect its input stream. (Any program compiled with this system can have its standard input and output redirected by typing <infile and >outfile on its command line when it is \*RUN. This is similar to, but distinct from, Arthur's redirection facility. It also indicates that the compiler is probably written in C and presumably compiled using itself - a common technique for porting compilers to new computers).

To make the compiler work with a symbol in this fashion it needs to be called via a Macro which does accept symbols within <> and can be made to pass them on to the compiler. It is the act of the CLI expanding the macro which causes the substitution so that it has already happened when the compiler is invoked. This would result in something like:

```
*SETMACRO ALIAS$FRED CC <C$File>
```

There is another way of dynamically controlling call lines which is also useful with the compiler. If a \*SET contains an expression of the form %\*0 then Arthur will substitute the rest of the command line into the translation. An example will make this more clear:

```
*SETMACRO ALIAS$FRED CAT %*0
```

This sets up FRED as an alias (a synonym or alternative) for CAT. Now the following two lines are equivalent:

```
*CAT $.LIBRARY
*FRED $.LIBRARY
```

This approach again works with the compiler when embedded in a macro. Thus we can have a line of the following form (the last bit is the important bit!):

```
*SET ALIAS$Comp ECHO Compiling %0
on|MTIME|MCC %*0
```

This sets up 'Comp' as a command to compile (the |M is a return ); if it is called as: \*Comp fred it expands to

```
*ECHO Compiling fred on
*TIME
*CC fred
```

The command ECHO simply prints the rest of the line, TIME prints the current time while CC, the C compile command, now has the correct parameter.

The difference between %\*0 and %0 is that the former selects all of the rest of the command line, %0 just selects one symbol (word); the use of %\*0 allows for a file name and some compiler optional parameters to be supplied to Comp.

This has solved the problem with C\$File in a more general fashion since Comp will take a file name or a variable (like <C\$File>) as a parameter, as well as any options required. Comp is used as a building block for the following lines which should now make sense (each command must be on one line without returns):

```
SET ALIAS$Compile ECHO Compiling
and Linking %0 on|MTIME|M CC
%*0 -link -arthur
SET ALIAS$Comp ECHO Compiling %0
on|MTIME|MCC %*0
SET ALIAS$Linkit ECHO Linking %0
on|MTIME|M LINK -file O.%0 -
library $.ARM.CLIB.O.ANSILIB,
$.ARM.CLIB.O.ARTHURLIB -IMAGE
P.%0 -ADFS -V %*1
SET ALIAS$Tidy ECHO Tidying disc|M
REMOVE O.%0|MREMOVE P.%0|M
COMPACT|MFREE|MMAF|M
SETMACRO ALIAS$C2 Comp <C$File>
SETMACRO ALIAS$C2L Compile <C$File>
SETMACRO Alias$Lc Echo Listing
C.<C$File> on|MTime|M|Mlist
C.<C$File>
SETMACRO Alias$Rc Echo Running
P.<C$File> on|MTime|MRUN
P.<C$File>
```

These define a suite of simple-to-type commands to drive a C session.

Comp and Compile provide the basis of C2 (compile) and C2L (compile and link) which operate on C\$File. Linkit will Link only (e.g. after C2).

Lc and Rc provide a shorthand to list and run the current file respectively.

Note the use of %\*1 in Linkit to pick up all of the parameters except the first (i.e. parameter 1 and all that follow). Linkit also uses the -v option which forces the linker to list what it is doing. Both the linker and compiler can be invoked with -help as their sole parameter to list all of their options; this offers a little more than the manual.

The Tidy command is born of experience. The compile and link cycle opens a number of files; this can fail if the disc starts to become fragmented. The Linker in particular then gives a terse 'binary error' message before giving up. Tidy clears out the old copies of the file and compacts the disc to give the compiler and linker the maximum disc space to work with. It does not assume C\$File because of the way in which I use it but there is no reason why it should not be modified to do so.

### Interacting with the Editor

When leaving the editor during a C session the usual aim is to compile (or even compile, link and run) as quickly as possible. This leads to a further command providing a single way of ending the editor, tidying the disc and invoking the compiler. Some of this will depend on the editor in use but the principles remain the same.

It is worth briefly considering the effect of \*QUIT. This appears to cause the current macro (defined by \*SETMACRO) to terminate immediately. Thus it is not possible to continue a macro beyond a Quit (or, presumably, an OS\_Exit – see PRM page 335). The CC command for the compiler does this, as does an exit from the 6502 Emulator or BASIC. However it does not affect an EXEC file so this can be used for the 'Done editing, go compile' command. An EXEC file is a text file of commands with its type set to &FFE (via \*SETTYPE Fred FFE).

The following is the content of the 'Qedit' file I use with the Pascal editor:

```
*|***** Leaving Edit *****
*quit
stamp c.<c$file>
tidy <c$file>
if (<c$action> AND 1) >0 then c2
if (<c$action> AND 2) >0 then linkit
    <c$file>
if (<c$action> AND 4) >0 then Rc
```

The first line is a comment while the second exits the 6502 Emulator (it could equally exit from BASIC). The Stamp command adds a timestamp to the file (not necessary if the editor in use does this automatically). The Tidy macro above is then used to housekeep the disc.

The actual compile/link/run cycle is controlled by the variable c\$action; this takes values of 1 to

compile, 3 to compile & link and 7 to compile, link & run. If the !Boot file sets c\$action to 1 its value can be increased manually as confidence increases.

It would be better if this file could be automated to carry out each step until one fails. Unfortunately the compiler does not set Sys\$ResultCode to anything but zero even when compilation errors occur. Are there any suggestions as to how else this might be simply achieved?

### Loose Ends

All of the preceding needs to be tied together via the !Boot file. It must include the various Set... commands, while Qedit must be in the Library directory with the compiler. The following lines are also useful in !Boot:

```
*Quit
RS423
FPE240
FX 202 48
SETMACRO CLI$PROMPT <SYS$TIME>:
    <C$File>*
```

The first is necessary if Arthur is configured to start up in BASIC, the second loads the RS423 patch (optional if not using the serial port) and the third the Floating Point Emulator. The FX turns off the Caps Lock since C is case-sensitive and mostly lower-case; again this assumes that the configuration is to Upper-case for BASIC. Finally, the Arthur prompt is set to show both the current time and the setting of C\$File. My !Boot also ECHOes a reminder to set C\$File before doing anything else.

The attentive reader may have noticed a certain obsession with times and dates on everything! This is intended to help the less organised among us (me) to find the latest listing etc and not spend time hunting bugs using the wrong material. Recommended.

### Conclusions

I have tried to show a practical approach to using C. In the process some of the power of the Arthur CLI has been exploited. The C system is well engineered and satisfying to use. The Arthur CLI less so, but still capable of mechanising routine tasks and thus speeding up the mundane jobs associated with using a compiler rather than an interpreter.

I hope that article this inspires others to experiment with both and to relay their experiences through the magazine. **A**



# ASCII, Printer Drivers and First Word Plus

Ian Nicholls

The Archive editor has been making a plea for articles which explain the workings of the Archimedes to those of his readers who are new to computing, or at least to the Acorn range of machines. Spurred on by his challenge, here is an article which tries to shed a little light on communicating with printers, with special reference to the word processing package First Word Plus.

Your copy of First Word Plus (FWP) has arrived, you have read the manual and you have managed to store something in the "doc" directory. Now you want to print it out. If you have an Epson or one of the other standard printers supported by FWP, you will be able to produce something on your printer that looks like what you typed in. If, like me, your printer is "largely" Epson-compatible then you have a problem. However, help may be at hand in the form of one of the additional "printer drivers" developed by Archive readers and available from Norwich Computer Services. If nobody has yet written one for your printer, or you are just curious about how printer drivers work, you will have to turn to Appendix B of the FWP manual.

The task of creating a printer driver is not for the faint-hearted, but at least it is not as bad with FWP as designing a printer driver routine for ArcWriter! The main purpose of this article is to describe how to produce a printer driver for use with FWP, using the STAR Gemini-10X as a typical "nearly Epson-compatible" printer. In this first part, I will go back to first principles and explain how microcomputers make printers write text, change fonts, draw graphics, change line spacing, etc. In the second part of the article, next month, I will also describe how to create, save and load alternative character sets for the Archimedes. Hopefully, the relevance of this last bit will emerge as the article unwinds!

## ASCII Codes - Communicating with Printers

Printer manufacturers vary considerably in the details they provide on this subject, and some manuals are still confusing arising from less than perfect translations from Japanese into English, and the need to make the instructions applicable to any computer. In fact, all communication with a printer is achieved by sending one of the 256 possible ASCII codes from the computer connected to it. The word ASCII stands for the phrase "American Standard Code for Information Interchange" and it describes the interpretation to be placed on each of these 256 values. The figure 256 arises from the fact that there are 256 different combinations of the 8 binary

digits (or bits), 0 and 1, which go to make up a byte: you make a printer work by sending a series of bytes to it. Until recently, the ASCII codes from 128 to 255 have had no standard interpretations: different manufacturers of computer hardware have assigned their own interpretations to them. However, on page 453 of the Archimedes User Guide, Acorn note that the International Standards Organisation (ISO) has now issued its specification ISO 8859 which contains two standard alphabets (Latin1 and Latin2) which assign specific characters to the ASCII codes from 160 to 255. The Arthur 1.2 OS ROM contains both of these alphabets, three others which are awaiting ISO approval (Latin3, Latin4 and Greek), and the Teletext and BFONT character sets.

ASCII Code	Corresponding characters
32 to 47	space ! " # \$ % & ' ( ) * + , - . /
48 to 57	0 1 2 3 4 5 6 7 8 9
58 to 64	; : < = > ? @
65 to 90	A B C ... Z
91 to 96	[ \ ] ^ _ `
97 to 122	a b c ... z
123 to 126	{ } ~
127	backspace plus delete.

The remaining codes from 0 to 31 have standard interpretations which hark back to teleprinter days with names like "BELL" for ASCII code 7!

To help with understanding the next section, you should load a BASIC program first; any will do, but preferably a short one! The program's listing on screen will consist of characters with ASCII codes between 32 and 126. If you have your printer switched on hold down the Ctrl key and press and release the B key: alternatively, type VDU2. If you now type LIST, your program will be listed on the printer as well as on the screen. To stop further text being sent to the printer when the listing has finished, you should press Ctrl and C, as you did earlier with Ctrl and B, or type VDU3.

To send output to the printer as well as to the screen, from within a BASIC program or from the keyboard, you must first send the ASCII code 2. If you only want output to go to the printer and not to the screen, then each character sent must have ASCII code 1 in front of it. To output a character on the Archimedes there are three methods: for ASCII code 2, for example, they are:-

<Ctrl-B>, VDU2 and PRINT CHR\$(2);

From now on we will use the middle method. If your

printer is a dot matrix type and you look again at the listing that you have just produced, you will probably be able to see the separate dots that make up each character. This is because most printers initialise themselves when they are switched on to use the Pica typeface. With this typeface you will have 10 characters to the inch and the separate dots that make up the characters will be clearly distinguishable. Now, with virtually every printer, sending ASCII code 15 to it will cause it to print with condensed characters (17 characters per inch). You can change into this mode easily by typing VDU2,1,15; if you now LIST the BASIC program again you should see the difference. Note the ASCII code 1 between the 2 and the 15 to ensure that the 15 code only goes to the printer and not to the screen.

### Escape Codes

Your printer is capable of much more than this, however, but how do you access its other features? The only way open is to send various ASCII codes to it, but almost all of the 256 codes cause the printer to print some alphanumeric character or other. The solution lies in the use of so-called "escape" codes. Escape is the name that has been given to ASCII code 27. If you send this code to the printer, it interprets the next few characters that are sent, not as something to print, but as an instruction to alter the linespacing, or change the font, or enter a graphics printing mode, etc. All printers use escape plus ASCII code 64 (the character @) to re-initialise, so if you now type VDU2,1,27,1,64 the printer will print in PICA text again.

Also, with most printers, the sequence of ASCII codes 27 83 1 will cause it to print in subscript mode: this is another example of using an escape sequence, the character 27 followed by one or more other characters. The command to send to access subscript mode is, therefore, VDU2,1,27,1,83,1,1 (note the interleaved ones again). Try listing your program once more: better still, first type in the command, VDU2,1,15 after the one above. This will give subscript and condensed modes together, which results in a very tiny but very neat typeface. However, the line spacing is now a little excessive! To reset the printer type in VDU2,1,27,1,64 again (subscript mode can be turned off separately with the sequence VDU2,1,27,1,84).

Your printer manual is likely to list the various escape sequences in one of two ways (or even both). For subscript mode, for example, it may give the command as: CHR\$(27) CHR\$(83) CHR\$(1), or as ESC S 1

The two forms are identical since the ASCII code for escape (ESC) is 27, and that for S is 83.

Some of the escape sequences require you to include a number that you determine yourself: an example of this is the alteration of the size of a linefeed. Your printer will probably have more than one way of doing this, but a command it is almost certain to have is one which enables you to specify the linefeed as a multiple of 1/72 of an inch. On the Star Gemini-10X this command is CHR\$(27) CHR\$(65) CHR\$(n) or ESC A n, where n is the number of 1/72 inch you wish the paper to move upwards when you send a linefeed command (ASCII code 10). To make the size of a linefeed 1/8 inch, n must be 9, so the command to the printer would be VDU2,1,27,1,65,1,9. You could try experimenting with different values for n to use with the combined condensed and subscript mode given above, if you want to conserve your listing paper!

### Printer Drivers

As with most word processors, First Word Plus offers many different features which take advantage of the capabilities of modern printers. But, of course, not all printers support all of these features. To accommodate these differences FWP provides you with a set of files, each one of which defines the facilities available when using a particular manufacturer's printer. If you have a printer for which one of these files (called printer drivers) is provided, or your printer is fully compatible with such a printer, then you can give the name of the appropriate driver to FWP when prompted, and you should have no further problems. As stated at the beginning of the article, one of its aims is to enable you to configure your own printer driver if none of the standard ones is suitable. When FWP is printing out your text and it comes to a command to print in, say, bold text, it looks in the printer driver "cfg" file currently loaded to see what sequence of ASCII codes it must send in order to cause the printer to move into the bold typeface.

### Configuring a Driver for a Particular Printer

I will use the STAR Gemini-10X (SG10) as the example printer: you will need your own printer's instruction manual with you for this section so that you can make a note of the appropriate commands you will need to enter. You should also be sitting in front of your Archimedes with the FWP system disc in the drive. Start up FWP by pressing <shift-break>. When the start-up screen appears you are presented with the list of files in the "doc" directory. Remove the system disc and insert the FWP utilities disc, and make FWP read the new disc by moving the mouse pointer onto the "quit" icon (the cross in the top left corner of the window) and click the left mouse button. The files on the OPEN FILE screen will change to the list of directories "bak, cfg, etc." each with



a right pointing arrow to the side. Move the pointer onto the "hex" directory and click the left mouse button. The contents of the hex directory will now be displayed: this is a list of printer driver hex files. Scroll through the list until one called "matrix" appears and then select it as above with the pointer and mouse. Now click on the "OK" panel and the basic matrix printer driver hex file will come up on screen.

You should now scroll past the introductory box (outlined with asterisks) and bring up the section headed "Printer name". At the bottom of this section there is a line which does not begin with an asterisk; it reads "Epson FX/RX (9-pin matrix)". You need to alter this to contain the name of your printer, using the normal facilities of FWP (which must not be in WP mode – it will not be if you have followed all the above start-up instructions). It is worth noting that in turning this hex file into the printer driver configuration file itself (which will be stored in the "cfg" directory), the FWP Install utility will ignore all lines beginning with an asterisk. So you can include as many comment lines as you like by starting each one with an asterisk.

### Configuration Variables

The next section of the hex file has the title "Configuration variables". There are six values to enter, and you will see them at the end of the section: at the moment they read "0,0,C,48,0,0". For a dot matrix printer the first variable must remain 0. For most such printers the second variable should be 0 as well. The only variation is where you need to print each pixel twice in graphics mode, so as to double the width (such as for the Atari SMM804). In this case the value is 2. For the SG10 the value is 0.

The third variable is not explained well in the FWP manual: you are asked for "the size of a linefeed in the printer's own height units"! In fact this value, and the one you enter for function number 25 in the "printer characteristics" section, are closely linked. This latter function is the one used to alter the size of a linefeed, and you will be entering the escape sequence details to cause the linefeed to be changed to  $n/72$  inch. Almost certainly the standard size of a linefeed on your printer (after switching it on) will be  $1/6$  inch. Now, if FWP has altered the linefeed size (say in the graphics mode), it will need to use function number 25 again to reset it to  $1/6$  inch. To do this  $n$  will have to be 12 (since  $1/6 = 12/72$ ), so the "size of a linefeed in the printer's own height units" is 12. The values you enter in the hex file all have to be in hexadecimal units, and 12 is C in hexadecimal, so the third configuration variable takes the value C. If your linefeed distance is set to be something other than  $1/6$

inch ( $1/8$  inch, say) or your printer only alters linespacing in, say,  $1/216$  inch increments, the above description should enable you to work out what value to enter instead of C.

The fourth variable is used by FWP to draw a box around a picture you have included in the text, to show the actual size of the picture when it is printed out. The value itself is the printer's graphics resolution in pixels per inch, and it is linked to function number 26 in the "printer characteristics" section. Function 26 is the sequence of commands used to make the printer enter graphics mode. Now many Epson printers have a wide variety of graphics modes with values 480, 576, 640, 720, 960, and 1920 pixels per line. The one used in the "matrix" file that we are modifying is mode 5, "plotter graphics" mode, which gives 576 pixels across the page. The width of a graphics print-out on the Epson FX/RX printers is 8 inches, so the number of pixels per inch in this mode is 72. This is 48 in hexadecimal and that is the value you have on screen for the fourth variable.

(If you don't know hex notation, and why should you?!, if you type, in BASIC, PRINT ~72, for example, it will display 48, the hex value.)

On my SG10, there is no mode with 576 pixels, the nearest one is the normal density mode with 480 pixels (or dots). Again, the width of a graphics print-out on the SG10 is 8 inches, so the number of pixels per inch is 60, or 3C in hexadecimal. When I come to defining function number 26 below, I must remember to use the sequence of commands for normal density. The fifth configuration variable only applies to daisywheel printers and so remains 0. The last variable is supposed to cause FWP to pause between pages if you are using separate sheets: you enter the value 1 if you want this feature and 0 otherwise. However, be warned that it does not work on earlier versions of FWP! If yours is such a version, then send your discs back to Acorn and they will replace them with updated versions with this bug removed.

### Printer Characteristics

This part of the "matrix" file consists of a series of entries in the form of a table – each entry consisting of two parts. The first is a hexadecimal number from 0 to 33, possibly preceded by an asterisk, otherwise followed by one or more hexadecimal numbers separated by commas. The second part of the entry begins with an asterisk followed by some explanatory text. The purpose of these entries is to list the control codes needed by your printer to activate each of its features (bold, underline, italic, etc.).

Before I proceed further I need to describe a piece of terminology used in this part of FWP: it is the idea of a



"placeholder". You will remember from the discussion above that some features on a printer require you to enter a number of your own choosing; an example I used was that of altering the size of a linefeed. The command to change the linefeed on my SG10 is 27 65 n, where n is the number of 1/72 inch for the linefeed. In the "printer characteristics" section, such a variable number is dealt with by using a "placeholder". What this means is that when FWP needs to use such a command, and it finds a placeholder in the sequence of bytes for the command, it will substitute a value of its own. To indicate that a value is a placeholder you put in a figure of 80(hex) or larger. FWP subtracts 80(hex) from the placeholder and adds the value to be output to it. This means that in almost every case the placeholder actually takes the value 80(hex). The manual quotes an example for a daisy-wheel printer where the command to alter the width of a character is the width+1: in this case, the placeholder would need to take the value 81(hex). I will now take each function in turn: note that you can omit functions, but you must keep those that are left in ascending order. (N.B. these comments relate to the STAR Gemini-10X printer)

- 0 omit – only used for daisy-wheel printers
- 1 carriage return and line-feed (these are standard values for all printers): the codes to enter are D,A
- 2,3 omit – only used for daisy-wheel printers
- 4 vertical tab to line – this is the sequence of codes to move vertically down to a particular line. On my printer, this facility is available, but when I entered the appropriate ASCII codes, all the output was interspersed with very large vertical gaps! It is not a vital command, and my advice is to omit it.
- 5 absolute horizontal tab – this is the horizontal equivalent to the last command: again it is not vital, and is best omitted.
- 6 draft bold on – I decided to use both emphasised and double-strike modes for bold (the former prints each character once and then again slightly displaced to the right, and the latter prints the character once, moves the paper up very slightly, and prints it again). The codes to enter are 1B,45,1B,47.
- 7 draft bold off – here it is necessary to turn both emphasised and double-strike modes off: the codes are 1B,46,1B,48.
- 8,9 NLQ bold on and off – my printer does not support NLQ, so I omitted these functions.
- A,B draft italic on and off – the codes for these are 1B,34 and 1B,35 respectively.
- C,D,E,F,10,11 these functions relate to NLQ italic,

- draft light and NLQ light modes: the SG10 does not support NLQ or light text, so they are omitted.
- 12,13 draft superscript on and off – 1B,53,0 and 1B,54.
- 14,15 NLQ superscript on and off – omitted.
- 16,17 draft subscript on and off – 1B,53,1 and 1B,54.
- 18,19 NLQ subscript on and off – omitted.
- 1A,1B draft underline on and off – 1B,2D,1 and 1B,2D,0.
- 1C,1D NLQ underline on and off – omitted.
- 1E formfeed – this is C (12 in decimal) for all printers.
- 1F horizontal initialisation – although this function is optional, since it is sent once only at the beginning of a document it is worth using it to reset the printer and set the pitch to Pica: the codes to do this are 1B,40,12.
- 20 vertical initialisation – again, this function is optional, and is sent once at the beginning of the document: I have used it to set the character set to that for the USA, the codes are 1B,37,0.
- 21 printer reset – as explained above, the codes for this are 1B,40.
- 22 backspace – this is a standard ASCII code and takes the value 8.
- 23 carriage return – the code is D.
- 24 form length in lines – the code you enter is sent at the beginning of the document and the length used is obtained from the one that you define in the Page layout dialogue box. If you select Layout from the main menu, a sub-menu appears with Page layout as the third item in the menu. Choosing this option produces a dialogue box which, amongst other things, allows you to specify the length of a page in lines (i.e. the number of lines of text per page, say 54) and the maximum number of lines per sheet (usually 66). The codes to enter are 1B,43,80; note the placeholder, 80, because FWP will need to enter a variable value here for the actual size of the form.
- 25 set linefeed distance – I made reference to this function earlier when I was describing the third "configuration variable". I have used the command to alter the linefeed to 1/72 inch; the codes are 1B,41,80. Again, there is a placeholder for the value, n, to be entered for a linefeed of n/72 inches.
- 26 print graphics – I referred to this function when discussing the fourth configuration variable above: function 26 is the one that FWP uses for a line containing graphics data. In deciding which graphics resolution to use for the fourth configuration variable, I chose the one giving 60 dots per inch: this is "normal-density graphics" on the SG10. The codes for this are 1B,4B.



- 27 This function is reserved; leave asterisk in place.
- 28 draft Pica – the codes to enter are those to turn off enlarged characters (1B,57,0) followed by those to print in Pica (1B,42,1).
- 29 NLQ Pica – this is omitted.
- 2A draft Elite – the codes to enter are those to turn off enlarged characters (1B,57,0) followed by those to print in Elite (1B,42,2).
- 2B NLQ Elite – this is omitted.
- 2C draft condensed – the codes to enter are those to turn off enlarged characters (1B,57,0) followed by those to print in condensed mode (1B,42,3).
- 2D NLQ condensed – this is omitted.
- 2E draft expanded – the codes to enter are those to print in Elite (1B,42,2) followed by those to print in enlarged mode (1B,57,1).
- 2F,30,31,32,33 these functions relate to NLQ expanded and black, magenta, cyan and yellow inks: the corresponding features are not available on the SG10, so I have omitted them.

### Character Translation Table

This last section of the printer configuration hex file consists of the series of values A0 to FF (decimal 160 to 255) followed by a series of hex values like those in the section above. In the earlier section on ASCII codes, I noted that Acorn have used "Latin 1" as the default alphabet on the Archimedes. This assigns a particular set of characters to the ASCII codes from 160 to 255. To see what these characters are in FWP, all you have to do is to reveal the "Font Table" at the bottom of the screen, if it is covered up by the text window (see page 34 of the FWP manual). The contents of this font table alter when you install different printer drivers; only those characters that you define in the character translation table appear in the font table.

If you now open your Archimedes User Guide at page 454, and your printer's instruction manual at the page that lists all the characters that it will print, you will find two things. Firstly, there will be some characters in the Latin 1 alphabet that your printer cannot reproduce and, secondly, there will be some characters in your printer's character set that are not contained in Latin 1. You may well also find that even if your printer can print one of the Latin 1 characters with an ASCII code between 160 and 255, the printer assigns a different ASCII code to it! There will, in addition, be some particular characters such as the Spanish upside down question mark (¿)(!¡), which are not in your printer's standard character set but which its manual lists in a so-called "international character set". I will now try to explain how this last section of the printer driver enables you to put all of your

printer's character set into the FWP font table and therefore into any document you produce with FWP.

You should now look back at your monitor screen which will be showing the heading "Translation Table (Epson Printers)". Following this title there are a few comment lines each beginning with an asterisk. The next line after this just contains the entry A0: the reason for this is that ASCII code A0 (160 in decimal) is not a printable character in Latin 1. However, the next ASCII code is; it is the Spanish letter "¡" and the entry on screen reads A1,1B,52,7,5B,1B,52,0. The A1 means that what follows is the set of ASCII codes that FWP must send to the printer to cause it to print out ASCII code A1 (decimal 161).

On the Epson FX/RX series of printers, the escape sequence 1B,52,7 makes the printer change to the Spanish character set. What this means is that the printer will still print out its full standard character set except that a small set of characters will be from the Spanish alphabet rather than the default alphabet chosen when the machine is switched on. The default value for ASCII code 5B (decimal 91) on these printers is the left square bracket ([); when the printer uses the Spanish character set ASCII code 5B produces the "¡". So, in order to make an Epson FX/RX printer print the "¡", we need to make it change to the Spanish character set, print ASCII code 5B, then switch back to the default character set. Looking back at the A1 entry in the Translation Table again, you should now be able to work out what it is doing:-

A1	ASCII code used in the Archimedes
1B,52,7	change to Spanish character set
5B	print character 5B (decimal 91)
1B,52,0	change back to default (USA) character set

There are quite a few characters in Latin 1 which your printer will print by using this method of changing between different international character sets. However, there are many other characters which look to be impossible to print: some of them are impossible, but a surprisingly large number can be printed using a little cunning! One example is probably on screen in front of you now: it is the underlined lower-case a (ª). The ASCII code for this character is AA (170). If you look at the FWP Translation Table entry for AA you will see the code sequence 61,8,5F. These codes are:-

61	a
8	backspace
5F	underline

These codes make the printer print a lower-case "a", backspace the printhead one character and then print an underline character. An even trickier one in the Epson

FX/RX printer driver is the character "e". The Latin 1 ASCII code for this is EB (decimal 235). If you scroll the text window down to this character you will see the following sequence of ASCII codes - 65,8,1B,52,1,7E,1B,52,0. These codes cause the printer to:-

- 65 print a lower-case "e"
- 8 backspace
- 1B,52,1 change to the French character set
- 7E print character 7E (decimal 126): this is " " in the French character set on Epson FX/RX
- 1B,52,0 change back to default (USA) character set

What you now have to do is to find which of the Latin 1 characters are capable of being printed on your printer, note down the sequence of ASCII "escape" and other codes that the printer needs to print them and then modify the standard Translation Table by entering these codes against the appropriate characters. If you just leave a blank against one of the Translation Table entries, then FWP will print out a space when it comes

across this character in your documents. If you omit a Table entry entirely the corresponding ASCII code will be sent straight to the printer without being translated: this could give undesirable results! You should check your manual to see whether any of the codes 160 to 255 do something other than print a character.

### Installing your Printer Driver

If you have been using this article to create a new printer driver, you have now reached the point where it has to be converted into a configuration file; at present it is just a hex file. First you must save the file in the usual way. To accomplish this conversion you need to use the 1stInstall program which is stored on your FWP utilities disc. The instructions on using 1stInstall are on pages 234 and 235 of the FWP manual and, since this article is already rather long, you should refer to the manual for the step-by-step instructions, and then try out your new "cfg" file with the "Test" file stored in the "docs" directory on the FWP system disc. **A**

## Image Processing from Holland

### Elaine Vaughan, Spacotech

A.I.M. is a public domain image processing package not originally intended for the Archimedes but, as it is written in 'C', it has been successfully transferred between a number of different computers during the course of its development. The software makes extensive use of the wimp environment, thus providing an excellent user interface that requires little instruction for anyone familiar with the desktop. The package does however need a multisync monitor as it operates entirely in mode 20.

Images are loaded from the selected filing system, or grabbed using the Watford digitiser, which has been anticipated by the software. The user is presented with three windows for image display and one window for commands. The provision of three windows is a good idea, as it allows two different images to be processed with the results displayed in the third window, a concept that would be familiar to machine code programmers aware of ARM's treble operand instructions. Disappointingly, each window only allows a maximum of 256 pixels square to be displayed, which means that it is not possible to handle high resolution images that might be expected from such a hardware setup.

There are no such shortcomings in the processing though. An extremely comprehensive set of

commands is available, neatly categorised into such operational groups as greyscale, cell and bit. There are several types of filters that can be applied to the image data along with various utility functions for histogram display etc. With all these operations available, there is an excellent facility for producing macro files that perform sequences of operations automatically. In practise, using data processing on images is a multi-step affair in which the user has his or her own favourite tricks based on try-it-and-see type methods therefore the facility to log keyboard input into macros is a welcome inclusion.

I had no difficulty in loading a picture received by the Spacotech weather satellite podule and was able to produce some interesting effects with the image. Also a number of demonstration sequences were included on the review disk along with sample images. All were impressive and demonstrations of noise cancellation, area estimation and pattern recognition proved that the package would be of great use to serious users wishing to apply analytical techniques to image data.

One point that I must make is that the software has no respect for the configuration of the host computer, as it is impossible to exit to an even ...Continued on page 42.



# Fact-File

Akom Ltd	51 Manor Lane, London, SE13 5QW. (01-852-4575)
Alpine Software	P.O.Box 25, Portadown, Craigavon, BT63 5UT. (0762-42510)
Blue Grey Software	28 Bobbitts Way, Wivenhoe, Essex, CO7 9NJ.
Cambridge Microsystems	19 Panton Street, Cambridge, CB2 1HL. (0223-66553)
CJE Micros	78 Brighton Road, Worthing, W Sussex, BN11 2EN. (0903-213361)
Clares Micro Supplies	98 Middlewich Road, Rudheath, Northwich, Cheshire, CW9 7DA. (0606-48511)
Colton Software	149-151 St Neots Road, Hardwick, Cambridge, CB3 7QJ. (0954-211472)
Computer Concepts	Gaddesden Place, Hemel Hempstead, Herts, HP2 6EX. (0442-63933)
Computerware	55 Romulus Court, Brentford Dock, Justin Close, Brentford, Middlesex.
Dabs Press	76 Gardner Road, Prestwich, Manchester, M25 7HU. (061-766-8423)
Dudley Micro Services	32 Osborne Road, Penn, Wolverhampton, WV4 4AY. (0902-342214/334315)
EMR Ltd	14 Mount Close, Wickford, Essex, SS11 8HG. (0702-335747)
GEM Electronics	17 Tandragee Road, Portadown, Craigavon, BT62 3BQ.
Ian Copestake Software	10 Frost Drive, Wirral, L61 4XL. (051-648-6287)
Intelligent Interfaces	43B Wood Street, Stratford-on-Avon, CV37 6JQ. (0789-415875)
Jansons	93 North Street, Stilton, Peterborough, PE7 3RR. (0733-244702)
Lingenuity	P.O.Box 10, Halesworth, Suffolk, IP19 0DX. (0986-85-476)
Minerva Systems	69 Sidwell Street, Exeter, EX4 6PH. (0392-437756)
Mitre Software	26 Creechurch Lane, London, EC3A 5BA. (01-283-4646)
Psientific Software	37 Cottesmore Road, Hessle, North Humberside, HU13 9JQ. (0482-649187)
P.Taylor Software	23 Limes Avenue, Alfreton, Derby, DE5 7DW. (0773-834893 or 768899)
RESOURCE	Exeter Road, Doncaster, DN2 4PY. (0302-63800/63784)
Richard Millican	10 Stokesay, Bidston, Birkenhead, L43 7PU
Robico Software	3 Fairland Close, Llantrisant, Mid Glamorgan, CF7 8QH. (0443-227354)
Silicon Vision Ltd	Signal House, Lyon Road, Harrow, Middlesex, HA1 2AG. (01-422-2274)
Solidisk Technology	87 Bournemouth Park Road, Southend-on-Sea, SS2 5JJ.
Spacetech	21 West Wools, Portland, Dorset, DT5 2EA. (0305-822753)
Superior Software	Regent House, Skinner Lane, Leeds, LS7 1AX. (0532-459453)
Texellence	2 Greenhill Road, Coleraine, N Ireland, BT51 3JE.
XOB	Balkeerie, Eassie by Forfar, Angus, DD8 1SR. (0307-84364)

**Norwich Computer Services 18 Mile End Road, Norwich, NR4 7QY. (0603-507057)**

# Archive

Subscription Magazine and Support Group for *Archimedes* Users

## Archive Magazine contains:

- News
- Reviews
- Hints and Tips – a major feature
- Articles for Beginners
- The Latest Technical Information
- Program Listings
- Free Small Ad's Section
- HELP – Requested and Offered
- Contact Box – to help you form common interest groups

## Eureka! – Bulletin Board

0603-250689 on 1200/75 or 300/300

- Very Latest News
- Down-load Software
- Mailbox Facilities
- Chat line

## Technical Help Service (£8 / year)

A telephone hot-line service for immediate help with your technical problems. Any member can send written enquiries, but for a fast response use the THS!

**Members' Discount:** 7.5% off software from Computer Concepts, Minerva, Clares, Acorn and others when purchased through Norwich Computer Services.

**Subscription:** 12 issues £14.50 (UK)  
Europe £20, Middle East £25,  
America / Africa £28, Elsewhere £30.  
Technical Help Service £8

N.B. All earlier issues have now been re-printed – you may back-date your subscription as far as issue 1 (October 1987) – to take advantage of this huge bank of information.

Archimedes is a trademark of Acorn Computers Ltd.

\* Please send copies of *Archive* magazine for one year starting from  
Volume \_\_\_\_ Issue \_\_\_\_

\* Please enrol me on the Technical Help Service for one year. (£8)  
I enclose a cheque for £ \_\_\_\_\_ payable to "Norwich Computer Services".

Name: \_\_\_\_\_

Address: \_\_\_\_\_

Postcode: \_\_\_\_\_

**Norwich Computer Services, 18 Mile End Road, Norwich, NR4 7QY**

Special two-year subscription starting from volume 1, issue 1 – £27